



TUGAS AKHIR - KI141502

**PENYELESAIAN PENJADWALAN NO-WAIT
FLOW SHOP MENGGUNAKAN METODE
METAHEURISTIK ALGORITMA GENETIKA DAN
TABU SEARCH**

**ARI MOHAMAD BARKHAH
NRP 5110100221**

**Dosen Pembimbing I
Ahmad Saikhu, S.Si., MT.**

**Dosen Pembimbing II
Rully Soelaiman, S.Kom., M.Kom.**

**Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



UNDERGRADUATE THESES - KI141502

SETTLEMENT SCHEDULING NO-WAIT FLOW SHOP USE THE METAHEURISTIC METHOD GENETIC ALGORITHMS AND TABU SEARCH

**ARI MOHAMAD BARKHAH
NRP 5110100221**

**First Advisor
Ahmad Saikhu, S.Si., MT.**

**Second Advisor
Rully Soelaiman, S.Kom., M.Kom.**

**Department of Informatics
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2015**

LEMBAR PENGESAHAN

PENYELESAIAN PENJADWALAN NO-WAIT FLOW SHOP MENGGUNAKAN METODE METAHEURISTIK ALGORITMA GENETIKA DAN TABU SEARCH

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Dasar dan Terapan Komputasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh:

**ARI MOHAMAD BARKHAH
NRP: 5110100221**

Disetujui oleh Pembimbing Tugas Akhir

1. **Ahmad Saikhu, S.Si., MT.**
(NIP. 197107182006041001)
(Pembimbing 1)
2. **Rully Soelaiman, S.Kom., M.Kom.**
(NIP. 197002131994021001)
(Pembimbing 2)



**SURABAYA
JULI, 2015**

PENYELESAIAN PENJADWALAN NO-WAIT FLOW SHOP MENGGUNAKAN METODE METAHEURISTIK ALGORITMA GENETIKA DAN TABU SEARCH

Nama Mahasiswa : ARI MOHAMAD BARKHAH
NRP : 5110100221
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Ahmad Saikhu, S.Si., MT.
Dosen Pembimbing 2 : Rully Soelaiman, S.Kom., M.Kom.

Abstrak

Penjadwalan no-wait flow shop (NWFS) adalah penjadwalan yang terdiri lebih dari satu pekerjaan dan mesin yang dilakukan secara berurutan dari mesin pertama hingga mesin terakhir dengan meminimalkan adanya waktu tunggu antar proses pada mesin. Berbagai industri sudah banyak menggunakan penjadwalan NWFS, diantaranya adalah industri besi, baja, dan pengalengan makanan.

Penelitian ini mengimplementasikan metode metaheuristik Algoritma Genetika dan Tabu Search (GA-TS) pada permasalahan penjadwalan NWFS. Metode TS dipilih karena metode GA memiliki kecenderungan terjebak pada local optima dan metode TS dapat meningkatkan kinerja metode GA dengan penurunan rata-rata hasil 4,45%.

Pada NWFS sering didapatkan penjadwalan yang tidak optimal. Hambatan sering terjadi karena urutan pekerjaan yang tidak optimal. Tugas akhir ini memberikan hasil waktu dan urutan pekerjaan yang terbaik. Dari solusi yang didapatkan, dapat diketahui bahwa metode GA-TS dapat menghasilkan solusi yang terbaik pada permasalahan penjadwalan NWFS.

Kata kunci: Penjadwalan, No-wait, Makespan, Algoritma Genetika, Tabu Search.

SETTLEMENT SCHEDULING NO-WAIT FLOW SHOP USE THE METAHEURISTIC METHOD GENETIC ALGORITHM AND TABU SEARCH

Student's Name : ARI MOHAMAD BARKHAH
Student's ID : 5110100221
Department : Teknik Informatika FTIF-ITS
First Advisor : Ahmad Saikhu, S.Si., MT.
Second Advisor : Rully Soelaiman, S.Kom., M.Kom.

Abstract

No-wait flow shop (NWFS) scheduling is a scheduling consist of more than one job and machine performed sequentially from the first machine to the last machine by minimizing the waiting time between processes on the machine. Many industries already use scheduling NWFS, including the iron, steel, and canning food.

The final project will implement a meta-heuristic method Genetic Algorithm and Tabu Search (GA-TS) on NWFS scheduling problems. TS method chosen for the GA method because it has a tendency to get stuck in local optima and TS can improve the performance of GA method with a decrease in average score at 4,45%.

NWFS often obtained unoptimal scheduling. Constrains often occur due to non-optimal job sequence thus much time wasted during the processing of the products. This final project will provide the best result time and work sequence. From the solution obtained, it can be seen that the GA-TS method can produce the best solution for the scheduling problem NWFS.

Keywords : Scheduling, No-wait, Makespan, Genetic Algorithm, Tabu Search.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul:

**“Penyelesaian Penjadwalan *No-Wait Flow Shop*
Menggunakan Metode Metaheuristik Algoritma Genetika
dan *Tabu Search*”.**

Selesainya penelitian ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Djuwita Sasih dan Ir. Mohamad Ramelan yang telah memberikan dukungan doa, moral, dan material yang tak terhingga kepada penulis sehingga penulis dapat menyelesaikan penelitian ini.
2. Bapak Rully Soelaiman, S.Kom., M.Kom., yang telah memberikan motivasi, nasehat, bimbingan, dan bantuan yang banyak kepada penulis selama masa perkuliahan di Teknik Informatika ITS.
3. Bapak Ahmad Saikhu, S.Si., MT., selaku dosen pembimbing I yang telah memberikan arahan, bantuan, dan perhatian sehingga penulis dapat menyelesaikan penelitian ini.
4. Segenap dosen dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di ITS.

Penulis menyadari bahwa penelitian ini masih memiliki banyak kekurangan sehingga dengan kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juli 2015
Ari Mohamad Barkhah

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak.....	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE SUMBER.....	xxi
DAFTAR NOTASI.....	xxiii
BAB 1 BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Permasalahan	2
1.4 Tujuan	3
1.5 Manfaat.....	3
1.6 Metodologi	3
1.6.1 Penyusunan Proposal	4
1.6.2 Studi Literatur	4
1.6.3 Implementasi Perangkat Lunak.....	4
1.6.4 Pengujian dan Evaluasi.....	4
1.6.5 Penyusunan Buku	5
1.7 Sistematika Penulisan Laporan	5
BAB 2 BAB II TINJAUAN PUSTAKA.....	7
2.1 Penjadwalan <i>No-Wait Flow Shop</i>	7
2.1.1 Fungsi Tujuan <i>No-Wait Flow Shop</i>	9
2.1.2 Fungsi Batasan <i>No-Wait Flow Shop</i>	9
2.1.3 Perhitungan <i>Makespan No-Wait Flow Shop</i>	10
2.1.4 Perhitungan Persentase Perbedaan <i>Makespan</i>	14
2.2 Pendekatan Algoritma Genetika Pada Permasalahan Penjadwalan <i>No-Wait Flow Shop</i>	14
2.2.1 Membangkitkan Kromosom	16
2.2.2 Kawin Silang atau <i>Crossover</i>	17
2.2.3 Mutasi	19

2.3	Pendekatan <i>Tabu Search</i> Pada Permasalahan Penjadwalan <i>No-Wait Flow Shop</i>	21
BAB 3 BAB III PERANCANGAN PERANGKAT LUNAK.23		
3.1	Variabel yang Digunakan Pada Perancangan Penyelesaian <i>No-Wait Flow Shop</i> Menggunakan Metode GA-TS	23
3.2	Perancangan Metode GA-TS Untuk Mendapatkan Urutan Pekerjaan Terbaik	29
3.2.1	Perancangan Pembangkitan Kromosom	29
3.2.2	Perancangan Pengambilan Sampel Kromosom Untuk Metode TS.....	30
3.2.3	Perancangan Proses Metode TS	31
3.2.4	Perancangan Proses GA	33
3.2.5	Perancangan Pengurutan Kromosom Berdasarkan <i>Makespan</i> Terbaik.....	36
3.3	Perancangan Metode GA-TS Untuk Mendapatkan Nilai <i>Makespan</i>	37
3.3.1	Perancangan Perhitungan Nilai <i>Makespan</i>	38
3.4	Perancangan Metode GA-TS Untuk Permasalahan <i>No- Wait Flow Shop</i> Secara Umum.....	40
BAB 4 BAB IV IMPLEMENTASI.....		47
4.1	Lingkungan Implementasi	47
4.2	Implementasi Metode GA-TS Untuk Mendapatkan Urutan Pekerjaan Terbaik.....	47
4.2.1	Implementasi Pembangkitan Kromosom	48
4.2.2	Implementasi Pengambilan Sampel Kromosom Untuk Metode TS.....	49
4.2.3	Implementasi Proses Metode TS.....	50
4.2.4	Implementasi Proses GA.....	52
4.2.5	Impelementasi Pengurutan Kromosom Berdasarkan <i>Makespan</i> Terbaik.....	57
4.3	Implementasi Metode GA-TS Untuk Mendapatkan Nilai <i>Makespan</i>	58
4.3.1	Implementasi Perhitungan Nilai <i>Makespan</i>	58
BAB 5 BAB V HASIL UJI COBA DAN EVALUASI		61

5.1	Lingkungan Pengujian.....	61
5.2	Data Pengujian	61
5.3	Uji Kinerja.....	62
5.3.1	Uji Kinerja Data Uji Coba Kategori Kecil.....	65
5.3.2	Uji Kinerja Data Uji Coba Kategori Besar	67
5.4	Uji Perbandingan Parameter Metode GA-TS.....	69
5.4.1	Uji Perbandingan Parameter Jumlah Populasi.....	70
5.4.2	Uji Perbandingan Parameter Nilai Awal Iterasi <i>Tabu Search</i>	71
5.4.3	Uji Perbandingan Parameter Batas Iterasi <i>Tabu Search</i>	73
5.4.4	Uji Perbandingan Parameter Kawin Silang	74
5.4.5	Uji Perbandingan Parameter Mutasi	75
5.5	Uji Perbandingan Metode GA Dengan Metode GA-TS.....	77
BAB 6	BAB VI KESIMPULAN DAN SARAN.....	81
6.1	Kesimpulan.....	81
6.2	Saran.....	82
DAFTAR PUSTAKA	83
LAMPIRAN A	85
LAMPIRAN B	91
LAMPIRAN C	107
LAMPIRAN D	123
BIODATA PENULIS	133

DAFTAR TABEL

Tabel 2.1 Contoh Data Penjadwalan NWFS.....	11
Tabel 2.2 Waktu Tunggu	11
Tabel 2.3 Contoh Bentuk Kromosom	17
Tabel 2.4 Proses Kawin Silang	19
Tabel 2.5 Proses Mutasi	20
Tabel 3.1 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 1)	23
Tabel 3.2 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 2)	24
Tabel 3.3 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 3)	25
Tabel 3.4 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 4)	26
Tabel 3.5 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 5)	27
Tabel 3.6 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 6)	28
Tabel 3.7 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 7)	29
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak	47
Tabel 5.1 Spesifikasi Lingkungan Pengujian.....	61
Tabel 5.2 Pengaturan Variabel Pada Data Uji Coba	63
Tabel 5.3 Data Uji Coba Kategori Kecil (Bagian 1)	63
Tabel 5.4 Data Uji Coba Kategori Kecil (Bagian 2)	64
Tabel 5.5 Data Uji Coba Kategori Besar	64
Tabel 5.6 Hasil Uji Coba Kategori Kecil	65

Tabel 5.7 Hasil Uji Coba Kategori Besar (Bagian 1)	67
Tabel 5.8 Hasil Uji Coba Kategori Besar (Bagian 2)	68
Tabel 5.9 Data Uji Coba Perbandingan Parameter	69
Tabel 5.10 Skenario Parameter Jumlah Populasi Yang Diuji Coba	70
Tabel 5.11 Perbandingan <i>Makespan</i> Pada Uji Perbandingan Parameter Jumlah Populasi (Bagian 1)	70
Tabel 5.12 Perbandingan <i>Makespan</i> Pada Uji Perbandingan Parameter Jumlah Populasi (Bagian 2)	71
Tabel 5.13 Skenario Parameter Nilai Awal Iterasi <i>Tabu Search</i> Yang Diuji Coba	72
Tabel 5.14 Perbandingan <i>Makespan</i> Pada Uji Perbandingan Parameter Nilai Awal Iterasi <i>Tabu Search</i>	72
Tabel 5.15 Skenario Parameter Batas Iterasi <i>Tabu Search</i> Yang Diuji Coba	73
Tabel 5.16 Perbandingan <i>Makespan</i> Pada Uji Perbandingan Parameter Batas Iterasi <i>Tabu Search</i>	73
Tabel 5.17 Skenario Parameter Kawin Silang Yang Diuji Coba	74
Tabel 5.18 Perbandingan <i>Makespan</i> Pada Uji Perbandingan Parameter Kawin Silang	75
Tabel 5.19 Skenario Parameter Mutasi Yang Diuji Coba	76
Tabel 5.20 Perbandingan <i>Makespan</i> Pada Uji Perbandingan Parameter Mutasi	76
Tabel 5.21 Hasil Uji Perbandingan Metode GA dengan Metode GA-TS (Bagian 1)	77
Tabel 5.22 Hasil Uji Perbandingan Metode GA dengan Metode GA-TS (Bagian 2)	78

DAFTAR NOTASI

C_{max}	Nilai <i>makespan</i> dari suatu urutan kerja
n	Jumlah pekerjaan pada <i>dataset</i> NWFS
m	Jumlah mesin pada <i>dataset</i> NWFS
i	Indeks pekerjaan
j	Indeks mesin
l	Indeks pekerjaan kedua
s_{ij}	Waktu mulai pada pekerjaan ke- i dan mesin ke- j
p_{ij}	Waktu proses pada pekerjaan ke- i dan mesin ke- j
F_{ij}	Waktu selesainya proses pada pekerjaan ke- i dan mesin ke- j
d_{il}	Waktu tunggu dari pekerjaan i ke pekerjaan l
k	Jumlah iterasi yang dilakukan
X_{ij}	Posisi pekerjaan pada kromosom pada pekerjaan ke- i dan mesin ke- j
cr	Parameter kawin silang
m	Parameter mutasi
c_1, c_2	Operator penentu posisi pada proses kawin silang
pm_1, pm_2	Operator penentu posisi pada proses mutasi
Q_1	Representasi dari dua operator pada kawin silang

Q_2	Representasi dari dua operator pada mutasi
A_i^k	Kromosom hasil kawin silang pada GA
B_i^k	Kromosom hasil mutasi pada GA
X_i^k	Posisi kromosom i pada iterasi k
TS	Kemungkinan penerimaan kromosom setelah mutasi pada metode TS
Y	Nilai acak yang dibangkitkan dengan rentang 0 hingga 1
I_0	Nilai iterasi awal pada metode TS
α	Koefisien <i>alpha</i> untuk penurunan iterasi
$I(k)$	Nilai iterasi pada metode TS saat iterasi ke- k
MP	Nilai <i>makespan</i> dari metode yang ingin dibandingkan
MCR	Nilai <i>makespan</i> dari metode penyedia data uji coba

DAFTAR GAMBAR

Gambar 2.1 Pola <i>Pure Flow Shop</i>	8
Gambar 2.2 Pola <i>No-Wait Flow Shop</i>	8
Gambar 2.3 <i>Directed Graph</i> NWFS	12
Gambar 2.4 <i>Gantt Chart</i> NWFS	14
Gambar 2.5 Diagram Alir Algoritma Genetika NWFS	15
Gambar 3.1 <i>Pseudocode</i> Pembangkitan Kromosom Awal	30
Gambar 3.2 <i>Pseudocode</i> Pengambilan Sampel Kromosom (Bagian 1)	30
Gambar 3.3 <i>Pseudocode</i> Pengambilan Sampel Kromosom (Bagian 2)	31
Gambar 3.4 <i>Pseudocode</i> Proses Metode TS (Bagian 1).....	32
Gambar 3.5 <i>Pseudocode</i> Proses Metode TS (Bagian 2).....	33
Gambar 3.6 <i>Pseudocode</i> Proses Kawin Silang pada GA (Bagian 1).....	34
Gambar 3.7 <i>Pseudocode</i> Proses Kawin Silang pada GA (Bagian 2).....	35
Gambar 3.8 <i>Pseudocode</i> Proses Mutasi pada GA (Bagian 1)	36
Gambar 3.9 Pengurutan Kromosom Berdasarkan <i>Makespan</i>	37
Gambar 3.10 <i>Pseudocode</i> Perhitungan Nilai <i>Makespan</i> (Bagian 1).....	38
Gambar 3.11 <i>Pseudocode</i> Perhitungan Nilai <i>Makespan</i> (Bagian 2).....	39
Gambar 3.12 Diagram alir proses Penyelesaian NWFS dengan Menggunakan Metode GA-TS Secara Umum	40
Gambar 3.13 <i>Pseudocode</i> Umum Metode GA-TS (Bagian 1).....	41
Gambar 3.14 <i>Pseudocode</i> Umum Metode GA-TS (Bagian 2).....	42
Gambar 3.15 <i>Pseudocode</i> Umum Metode GA-TS (Bagian 3).....	43
Gambar 3.16 <i>Pseudocode</i> Umum Metode GA-TS (Bagian 4).....	44
Gambar 3.17 <i>Pseudocode</i> Umum Metode GA-TS (Bagian 5).....	45
Gambar 5.1 Keterangan Atribut Pada <i>Gantt Chart</i>	66
Gambar 5.2 <i>Gantt Chart car1.csv</i> (Bagian 1).....	66
Gambar 5.3 <i>Gantt Chart car1.csv</i> (Bagian 2).....	66

Gambar 5.4 Gantt Chart car1.csv (Bagian 3).....67

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Pembangkitan Kromosom	.48
Kode Sumber 4.2 Implementasi Pengambilan Sampel Kromosom Untuk Metode TS (Bagian 1)49
Kode Sumber 4.3 Implementasi Pengambilan Sampel Kromosom Untuk Metode TS (Bagian 2)50
Kode Sumber 4.4 Implementasi Proses Metode TS (Bagian 1)50
Kode Sumber 4.5 Implementasi Proses Metode TS (Bagian 2)51
Kode Sumber 4.6 Implementasi Proses Metode TS (Bagian 3)52
Kode Sumber 4.7 Proses Kawin Silang Pada Fungsi <i>main.m</i>	53
Kode Sumber 4.8 Implementasi Proses Kawin Silang Pada Fungsi <i>crossover.m</i> (Bagian 1)54
Kode Sumber 4.9 Implementasi Proses Kawin Silang Pada Fungsi <i>crossover.m</i> (Bagian 2)55
Kode Sumber 4.10 Implementasi Proses Irisan Pada Fungsi <i>irisan.m</i> (Bagian 1)55
Kode Sumber 4.11 Implementasi Proses Irisan Pada Fungsi <i>irisan.m</i> (Bagian 2)56
Kode Sumber 4.12 Implementasi Proses Mutasi (Bagian 1)	.56
Kode Sumber 4.13 Implementasi Proses Mutasi (Bagian 2)	.57
Kode Sumber 4.14 Implementasi Pengurutan Kromosom Berdasarkan <i>Makespan</i> Terbaik57
Kode Sumber 4.15 Implementasi Perhitungan Nilai <i>Makespan</i> (Bagian 1)58
Kode Sumber 4.16 Implementasi Perhitungan Nilai <i>Makespan</i> (Bagian 2)59
Kode Sumber 4.17 Implementasi Perhitungan Nilai <i>Makespan</i> (Bagian 3)60

(Halaman sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penjadwalan *no-wait flow shop* (NWFS) adalah penjadwalan yang terdiri dari n pekerjaan dan m mesin di mana pekerjaan tersebut dilakukan berurutan dari mesin pertama hingga mesin terakhir dengan meminimalkan waktu tunggu. Penjadwalan NWFS diciptakan karena terdapat masalah pada bidang industri di mana terdapat produk-produk pada industri yang tidak tahan lama apabila harus menunggu lama dari satu mesin ke mesin berikutnya untuk diproses. Produk harus menunggu lama untuk diproses karena adanya penumpukan pekerjaan pada satu mesin sehingga membuat panjang antrean pada mesin tersebut. Oleh sebab itu agar kemungkinan untuk terjadinya penumpukan pekerjaan kecil pada satu atau lebih mesin, urutan pekerjaan mesin harus dibuat seoptimal mungkin. NWFS mengoptimalkan kinerja aliran penjadwalan sehingga tidak membiarkan pekerjaan yang menunggu untuk diproses dari satu mesin ke mesin berikutnya. Kriteria urutan penjadwalan yang baik adalah tidak terdapat dua atau lebih pekerjaan menumpuk pada satu atau lebih mesin.

Salah satu industri yang membutuhkan penjadwalan NWFS adalah industri tambang. Industri tambang mengekstraksi bijih mentah menjadi bahan jadi tambang. Bahan jadi tambang biasanya berbentuk batangan atau plat. Pada saat proses ekstraksi terjadi, bijih harus dalam keadaan panas. Apabila bijih menjadi dingin, maka proses ekstraksi akan gagal. Oleh sebab itu, penjadwalan NWFS dibutuhkan karena dapat meminimalkan waktu tunggu bijih saat proses ekstraksi dan menghindarkan bijih menjadi dingin.

Tugas akhir ini mengimplementasikan metode metaheuristik *Genetic Algorithm* dan *Tabu Search* (GA-TS). Definisi dari metaheuristik adalah proses iterasi dari metode heuristik dengan cara mengkombinasikan konsep yang berbeda

untuk mengeksplorasi dan mengeksploitasi ruang pencarian [1]. Metaheuristik pada penelitian terdiri dari penggabungan dua proses metode, yaitu metode GA dan metode TS. Hal yang membedakan penggunaan metaheuristik dengan heuristik adalah apabila menggunakan metode heuristik, maka tidak ada penggabungan dua proses tersebut.

Metode GA dipilih karena dapat memberikan hasil yang optimal dan sama berkualitasnya dengan metode heuristik lainnya. Metode TS dipilih untuk mengoptimalkan hasil pencarian yang sebelumnya telah dihasilkan oleh metode GA dan menghindari pencarian solusi yang terjebak pada nilai *local minima* [2]. Hasil yang diharapkan adalah ditemukannya urutan penjadwalan yang optimal melalui metode metaheuristik GA-TS pada permasalahan NWFS dengan meminimalkan waktu tunggu pada pekerjaan saat pemrosesan terjadi.

1.2 Rumusan Masalah

Penelitian ini mengangkat beberapa rumusan masalah sebagai berikut:

1. Perancangan desain algoritma GA-TS untuk menyelesaikan permasalahan penjadwalan NWFS.
2. Mengimplementasikan metode GA-TS untuk menyelesaikan permasalahan penjadwalan NWFS.
3. Mengimplementasikan metode NWFS pada metode *flow shop*.

1.3 Batasan Permasalahan

Permasalahan yang dibahas pada penelitian ini memiliki batasan sebagai berikut:

1. Jenis penjadwalan *flow shop* yang digunakan adalah penjadwalan NWFS.
2. Algoritma yang digunakan adalah metode algoritma genetika yang ditambah dengan metode metaheuristik *Tabu Search*.

3. Data uji coba didapat dari uji coba yang dilakukan oleh Carlier [3] dan Reeves [4] .
4. Metode algoritma GA-TS membandingkan nilai *makespan* yang didapatkan oleh metode algoritma dari Carlier, Reeves, dan metode Algoritma Genetika dari Chaudry & Khan [5].
5. Metode algoritma GA-TS membandingkan lima parameter, yaitu parameter jumlah populasi, kawin silang, mutasi, nilai awal iterasi *Tabu Search*, dan batas iterasi *Tabu Search*.

1.4 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Memahami perancangan desain algoritma GA-TS untuk menyelesaikan permasalahan penjadwalan NWFS.
2. Mengimplementasikan metode GA-TS untuk menyelesaikan permasalahan penjadwalan NWFS.
3. Mengimplementasikan metode NWFS pada metode *flow shop* yang diselesaikan dengan metode GA-TS.
4. Mengevaluasi kinerja dari metode NWFS pada metode *flow shop* yang diselesaikan dengan metode GA-TS.

1.5 Manfaat

Penelitian ini dikerjakan agar dapat memberikan manfaat pada bidang pabrik industri dalam menentukan penjadwalan NWFS pada lebih dari satu mesin menggunakan metode metaheuristik GA-TS dengan meminimalkan waktu tunggu pada pekerjaan yang diproses.

1.6 Metodologi

Pembuatan penelitian ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal

Tahap awal penelitian ini adalah menyusun proposal penelitian. Pada proposal, diajukan gagasan untuk membantu menyelesaikan masalah menentukan waktu yang optimal pada penjadwalan NWFS dengan menggunakan metode metaheuristik GA-TS. Terdapat juga cara perhitungan untuk menentukan waktu yang optimal pada suatu urutan kerja yang sesuai dengan kriteria NWFS.

1.6.2 Studi Literatur

Pada tahap ini dilakukan untuk mencari informasi dan studi literatur apa saja yang dapat dijadikan referensi untuk membantu pengerjaan penelitian ini. Informasi didapatkan dari buku dan literatur yang berhubungan dengan metode yang digunakan. Informasi yang dicari adalah konsep permasalahan NWFS, konsep penggunaan *Gantt Chart* pada NWFS, metode algoritma genetika, dan metode metaheuristik *tabu search*.

1.6.3 Implementasi Perangkat Lunak

Implementasi merupakan tahap untuk membangun metode-metode yang sudah diajukan pada proposal penelitian. Untuk membangun algoritma yang telah dirancang sebelumnya, maka dilakukan implementasi dengan menggunakan suatu perangkat lunak sesuai dengan konsep yang telah didapatkan pada studi literatur.

1.6.4 Pengujian dan Evaluasi

Pada tahap ini algoritma yang telah disusun diuji coba dengan menggunakan data uji coba yang ada. Data uji coba tersebut diuji coba dengan menggunakan suatu perangkat lunak. Hasil pada uji coba tersebut dievaluasi mengenai kekurangan-kekurangan yang ada.

1.6.5 Penyusunan Buku

Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan penelitian yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan penelitian adalah sebagai berikut:

1. Bab I. Pendahuluan

Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan penelitian.

2. Bab II. Tinjauan Pustaka

Bab ini berisi penjelasan secara detil mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan penelitian. Penjelasan mengenai fungsi tujuan dan fungsi batasan pada permasalahan NWFS dijelaskan pada model sistematis yang juga berada pada bab ini.

3. Bab III. Perancangan Perangkat Lunak

Bab ini berisi pembahasan mengenai perancangan dari metode metaheuristik GA-TS yang digunakan untuk menyelesaikan permasalahan NWFS.

4. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari metode GA-TS untuk permasalahan NWFS. Kode sumber tersebut bertujuan untuk membangun sebuah program berdasarkan perancangan yang telah dilakukan sebelumnya.

5. Bab V. Hasil Uji Coba dan Evaluasi

Bab ini berisikan hasil uji coba dari metode GA-TS untuk permasalahan NWFS yang sudah diimplementasikan pada kode sumber. Uji coba tersebut dibandingkan dengan beberapa metode yang sudah ada sebelumnya dan sudah

diketahui hasilnya. Evaluasi dilakukan untuk mengetahui kelebihan dan kekurangan dari metode GA-TS dibandingkan dengan metode-metode yang sudah ada.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan penelitian, dan saran untuk pengembangan solusi ke depannya.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam penelitian.

8. Lampiran

Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode sumber program secara keseluruhan.

BAB II

TINJAUAN PUSTAKA

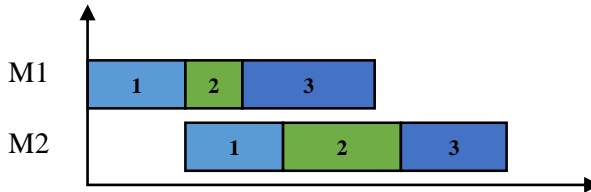
Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam penelitian. Teori-teori tersebut diantaranya adalah penjadwalan *no-wait flow shop*, metode Algoritma Genetika, dan metode *Tabu Search*.

2.1 Penjadwalan No-Wait Flow Shop

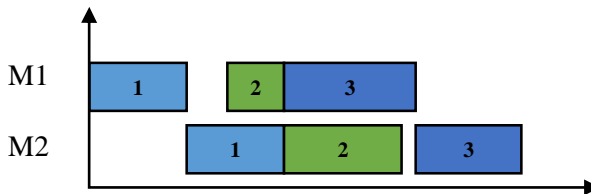
Penjadwalan *no-wait flow shop* (NWFS) adalah penjadwalan suatu urutan kerja di mana setiap pekerjaan harus diproses dari mesin pertama hingga terakhir secara berurutan dengan meminimalkan waktu tunggu. Penjadwalan NWFS termasuk salah satu dari jenis-jenis *flow shop* yang ada. Perbedaan *pure flow shop* dengan NWFS adalah waktu tunggu pada NWFS harus seminimal mungkin saat perpindahan dari satu mesin ke mesin berikutnya sedangkan *pure flow shop* tidak mempermasalahkan adanya waktu tunggu tersebut. Pada Gambar 2.1 dan Gambar 2.2 ditunjukkan perbedaan dari kedua *flow shop* tersebut dengan menggunakan *Gantt Chart*.

Pada *pure flow shop* pekerjaan 1, 2, dan 3 dapat langsung diproses oleh mesin pertama secara berurutan. Pada saat pekerjaan 2 selesai diproses oleh mesin pertama, pekerjaan 2 tidak dapat diproses oleh mesin kedua karena mesin kedua masih memproses pekerjaan 1. Hal tersebut menyebabkan terdapatnya waktu tunggu pada pekerjaan 2. Hal yang sama terdapat pada pekerjaan 3, pekerjaan 3 tidak dapat langsung diproses setelah dari mesin pertama karena mesin kedua masih memproses pekerjaan 2. Pada NWFS, pemrosesan pekerjaan 2 oleh mesin pertama ditunda. Penundaan tersebut dilakukan agar pekerjaan 2 dapat diproses dari mesin pertama ke mesin kedua secara langsung. Pekerjaan 3 dapat diproses langsung setelah pekerjaan 2 selesai diproses pada mesin pertama. Hal tersebut disebabkan karena pekerjaan 3 dapat

dipastikan tidak perlu menunggu jika telah selesai pada mesin pertama.



Gambar 2.1 Pola *Pure Flow Shop*



Gambar 2.2 Pola *No-Wait Flow Shop*

Pada penjadwalan NWFS terdapat n pekerjaan dan m mesin dengan ketentuan $M \geq 2$. Permasalahan yang diselesaikan pada penjadwalan NWFS adalah meminimalisasi total waktu tempuh pada seluruh pekerjaan dari mesin pertama hingga terakhir atau biasa disebut dengan *makespan*. Penjadwalan NWFS juga memiliki beberapa batasan permasalahan, diantaranya adalah semua mesin selalu tersedia sejak waktu 0, waktu pengolahan dan jumlah mesin yang dibutuhkan sudah diketahui dari awal, waktu perpindahan antar mesin dijadikan satu pada waktu proses, setiap mesin tidak dapat memproses lebih dari satu pekerjaan, mesin tidak dapat diinterupsi, setiap pekerjaan memiliki waktu prioritas yang sama, dan setiap pekerjaan mengikuti sesuai urutan yang sudah ditentukan. Untuk mendapatkan nilai *makespan* yang memenuhi kriteria NWFS dibahas lebih lanjut pada subbab 2.1.3.

Cara yang ditempuh agar mendapatkan *makespan* terbaik adalah mencari urutan pekerjaan yang terbaik diantara beberapa

kemungkinan yang ada. Permasalahan NWFS dapat diterjemahkan menjadi pemodelan matematis seperti persamaan-persamaan yang terdapat pada subbab 2.1.1 dan 2.1.2. Subbab 2.1.1 dan subbab 2.1.2 menjelaskan mengenai fungsi tujuan dan fungsi batasan pada NWFS. Penjelasan mengenai notasi-notasi yang digunakan dapat dilihat pada Daftar Notasi.

2.1.1 Fungsi Tujuan *No-Wait Flow Shop*

Subbab ini membahas mengenai tujuan dari NWFS. Tujuan dari NWFS adalah meminimalisasi waktu *makespan* seperti yang ditunjukkan oleh Persamaan 2.1. Nilai *makespan* adalah nilai waktu pada saat pekerjaan terakhir selesai diproses oleh mesin terakhir seperti yang ditunjukkan oleh Persamaan 2.2. Perhitungan untuk mendapatkan nilai *makespan* secara umum adalah dengan cara menambahkan waktu mulai pekerjaan i pada mesin j dengan waktu proses pekerjaan i pada mesin j seperti yang ditunjukkan oleh Persamaan 2.3. Sedangkan perhitungan untuk menentukan waktu dimulainya proses pada mesin pertama oleh suatu pekerjaan ditunjukkan oleh Persamaan 2.4.

$$\text{Minimalisasi } C_{max} \quad (2.1)$$

$$C_{max} = F_{ij} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.2)$$

$$F_{ij} = S_{ij} + P_{ij} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.3)$$

$$F_{ij} = S_{ij} + P_{ij} + d_{il} \quad i, l = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.4)$$

2.1.2 Fungsi Batasan *No-Wait Flow Shop*

Penjadwalan NWFS memiliki batasan-batasan yang harus dipenuhi agar memenuhi kriteria NWFS itu sendiri. Apabila terdapat satu atau lebih batasan yang tidak dipenuhi, maka penjadwalan tersebut tidak dapat dikatakan NWFS. Batasan-batasan tersebut antara lain adalah setiap pekerjaan harus

dikerjakan oleh satu atau lebih mesin, setiap pekerjaan harus diproses hingga selesai oleh satu mesin sebelum dapat berpindah ke mesin berikutnya sesuai urutannya, setiap mesin harus menyelesaikan pekerjaan yang sedang dikerjakan terlebih dahulu sebelum dapat memproses pekerjaan lainnya, dan setiap pekerjaan harus diproses secara berkelanjutan dari mesin pertama hingga mesin terakhir dengan meminimalkan waktu tunggu. Batasan-batasan tersebut dimodelkan kedalam suatu model sistematis. Model sistematis tersebut ditunjukkan secara berurutan oleh Persamaan 2.5, Persamaan 2.6, Persamaan 2.7, dan Persamaan 2.8.

$$1 \leq \sum_{j=1}^m x_{ij} \leq m \quad i = 1, 2, \quad (2.5)$$

$$F_{ij} \leq S_{i(j+1)} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.6)$$

$$S_{yj} \leq F_{xj} \quad \text{Semua job}(x, y) \quad i = 1, 2, \dots, m \quad (2.7)$$

$$F_{ij} \geq F_{(i-1)j} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.8)$$

2.1.3 Perhitungan *Makespan No-Wait Flow Shop*

Subbab ini menjelaskan mengenai detail dan metode yang digunakan untuk perhitungan *makespan* pada NWFS. Metode perhitungan *makespan* pada NWFS adalah dengan menggunakan *directed graph* sedangkan untuk melakukan pembuktian bahwa penjadwalan tersebut memenuhi kriteria NWFS dapat dibuktikan dengan menggunakan *gantti chart*.

Directed graph adalah kumpulan dari suatu simpul yang saling terhubung oleh suatu sisi. *Directed graph* digunakan karena salah satu metode yang mudah digunakan untuk menerjemahkan perhitungan dari NWFS. Pemodelan *directed graph* dapat mudah dipahami karena pemodelan tersebut sederhana dan memiliki arah panah untuk mengikuti alur perhitungannya. Pada perhitungan *makespan* dibutuhkan waktu tunggu yang seminimal mungkin.

Waktu tunggu yang dimaksud adalah waktu kosong diantara pekerjaan satu dengan setelahnya pada mesin pertama. Waktu tunggu ini sangat penting karena mempengaruhi waktu yang dilalui oleh suatu pekerjaan dari awal hingga akhir. Waktu tunggu antar pekerjaan didapat dari perhitungan yang dilakukan oleh Reddy & Ramamoorthy [6]. Perhitungan tersebut ditunjukkan oleh Persamaan 2.9.

$$d_{il} = \max_k \left(\sum_{i=2}^k t_{ij} - \sum_{i=1}^{k-1} t_{lj}, \quad 0 \right) \quad (2.9)$$

Agar lebih mudah untuk memahami perhitungan *makespan*, diberikan suatu contoh kasus penjadwalan NWFS seperti yang ditunjukkan oleh Tabel 2.1. Tabel 2.1 menunjukkan terdapat penjadwalan *flow shop* dengan spesifikasi tiga mesin dan tiga pekerjaan.

Tabel 2.1 Contoh Data Penjadwalan NWFS

Pekerjaan	Waktu Proses Pada Mesin (t)		
	1	2	3
1	2	4	4
2	3	5	1
3	2	4	3

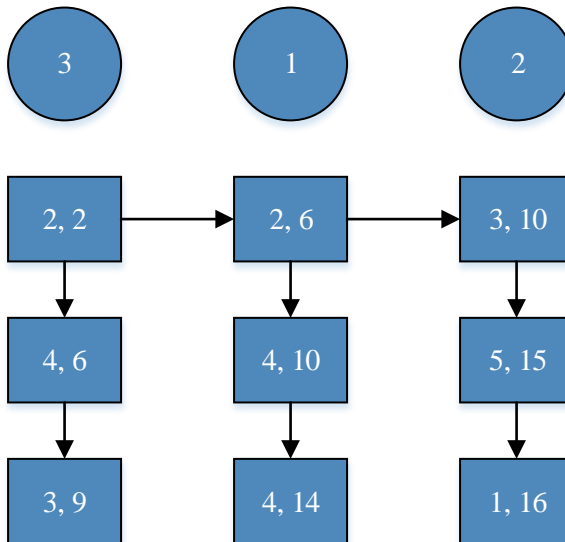
Tabel 2.2 Waktu Tunggu

Waktu Tunggu		Pekerjaan (t)		
		1	2	3
Pekerjaan (t)	1	-	1	2
	2	3	-	3
	3	2	1	-

Dari Tabel 2.1 dapat diambil asumsi bahwa urutan pekerjaan 3-1-2 adalah yang terbaik. Waktu tunggu dihitung

dengan menggunakan perhitungan pada Persamaan 2.9. Waktu tunggu tersebut ditunjukkan oleh Tabel 2.2.

Tabel 2.2 menunjukkan waktu tunggu antar pekerjaan. Pada urutan pekerjaan 3-1-2, pekerjaan 1 harus menunggu selama $2t$ sebelum dapat memulai pemrosesan pada mesin pertama. Begitu juga dengan pekerjaan 2 yang harus menunggu selama $1t$ agar dapat memproses pekerjaannya pada mesin pertama.



Gambar 2.3 Directed Graph NWFS

Perhitungan untuk mencari nilai *makespan* setelah diketahui waktu tunggu adalah dengan cara melakukan perhitungan dengan metode *directed graph* seperti yang ditunjukkan pada Gambar 2.3. Pada gambar tersebut, bentuk lingkaran menunjukkan urutan pekerjaan sementara bentuk persegi panjang menunjukkan waktu proses dan waktu selesai pekerjaan i pada mesin j . Kolom menunjukkan pekerjaan sedangkan baris menunjukkan mesin.

Pada Gambar 2.3 terdapat arah panah yang berasal dari mesin j menuju ke $j+1$ dan arah panah yang berasal dari pekerjaan

i menuju ke $i+1$. Arah panah tersebut menunjukkan adanya penambahan waktu pada saat pemrosesan. Untuk memastikan semua pemrosesan memenuhi batasan-batasan yang ada, maka dibuat tiga kondisi perhitungan.

Kondisi yang pertama adalah jika $j = 1$ dan $i > 1$, maka $S_{ij} = F_{(i-1)j} + d_{il}$. Kondisi tersebut menggunakan Persamaan 2.10 sebagai adaptasi dari fungsi tujuan pada Persamaan 2.4.

$$F_{ij} = F_{i(j-1)} + P_{ij} + d_{il} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.10)$$

Kondisi yang kedua untuk menghitung F_{ij} adalah jika nilai $F_{i(j-1)} \geq F_{(i-1)j}$, maka $S_{ij} = F_{i(j-1)}$. Kondisi tersebut menggunakan Persamaan 2.11 sebagai adaptasi dari fungsi tujuan pada Persamaan 2.3.

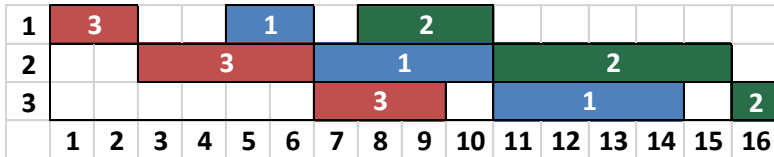
$$F_{ij} = F_{i(j-1)} + P_{ij} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.11)$$

Kondisi yang ketiga untuk menghitung F_{ij} adalah jika nilai $F_{(i-1)j} > F_{i(j-1)}$, maka $S_{ij} = F_{(i-1)j}$. Kondisi tersebut menggunakan Persamaan 2.12 sebagai adaptasi dari fungsi tujuan pada Persamaan 2.3.

$$F_{ij} = F_{(i-1)j} + P_{ij} \quad i = 1, 2, \dots, n \quad j = 1, 2, \dots, m \quad (2.12)$$

Nilai *makespan* dari Gambar 2.3 adalah F_{nm} . Untuk melakukan pembuktian apakah nilai *makespan* tersebut benar dan memenuhi kriteria NWFS, maka digunakan suatu metode yang bernama *Gantt Chart*. Metode ini sederhana, *robust*, dan sering digunakan untuk merencanakan dan memonitor aktivitas proyek. Metode ini sering digunakan pada *flow shop* jenis apapun untuk melakukan pembuktian apakah nilai *makespan* dan metode yang dijalankan valid atau tidak. Pembuktian keselarasan hasil

penggunaan *directed graph* pada metode NWFS ditunjukkan oleh Gambar 2.4.



Gambar 2.4 Gantt Chart NWFS

Dapat dilihat pada Gambar 2.4 bahwa *makespan* dari *directed graph* dan *gantt chart* menghasilkan nilai yang sama. Selain itu, batasan-batasan yang terdapat pada NWFS juga terpenuhi. Pekerjaan 1 memundurkan waktu proses pada mesin 1 karena waktu selesai pada mesin 1 kurang dari waktu selesai pekerjaan 3 pada mesin 2. Begitu juga yang terjadi dengan pekerjaan 2 yang memundurkan waktu proses pada mesin 1.

2.1.4 Perhitungan Persentase Perbedaan *Makespan*

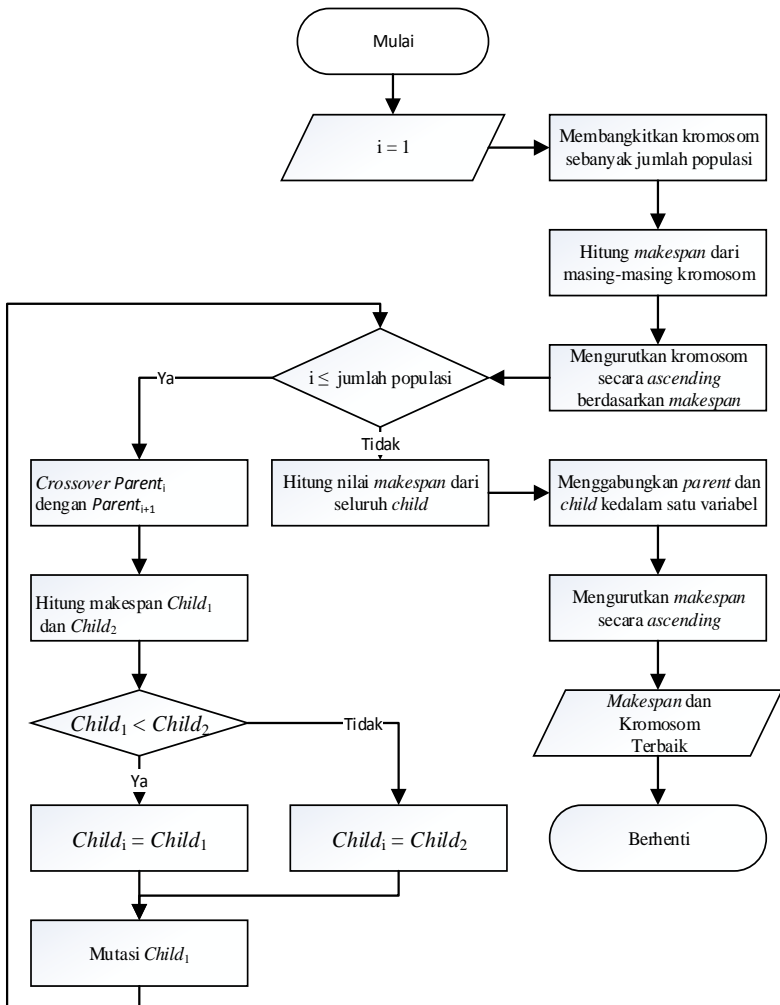
Pada subbab ini menjelaskan mengenai perhitungan perbedaan nilai *makespan*. Nilai *makespan* yang dibandingkan adalah nilai *makespan* dari penyedia data uji coba, yaitu Carlier dan Reeves dengan *makespan* yang ingin dibandingkan. Perhitungan perbedaan nilai *makespan* tersebut dijelaskan pada Persamaan 2.13. Penggunaan persentase tersebut dapat dilihat lebih lanjut pada Bab 5 mengenai hasil uji coba dan evaluasi.

$$\frac{MP - MCR}{MCR} \times 100\% \quad (2.13)$$

2.2 Pendekatan Algoritma Genetika Pada Permasalahan Penjadwalan *No-Wait Flow Shop*

Algoritma genetika (GA) adalah salah satu metode pencarian heuristik yang meniru dari proses seleksi alam dan

berdasarkan pada prinsip-prinsip genetika. GA termasuk dalam kelompok *Evolutionary Algorithm* (EA).



Gambar 2.5 Diagram Alir Algoritma Genetika NWFS

EA adalah metode penyelesaian untuk masalah optimasi dengan menggunakan teknik meniru dari proses seleksi alam, seperti mutasi, seleksi, dan kawin silang. Prosedur pencarian di dalam GA hanya didasarkan pada nilai fungsi tujuan dan tidak menggunakan nilai *gradient* atau teknik kalkulus. Alur pencarian GA dibagi dalam enam tahap, tahap pertama adalah membangkitkan populasi awal secara acak sesuai ukuran populasi yang diinginkan. Tahap kedua adalah mengevaluasi nilai setiap kromosom di dalam populasi awal dengan menggunakan fungsi *fitness*. Tahap ketiga adalah menyeleksi kromosom terbaik untuk disalin dalam jumlah tertentu untuk menggantikan kromosom lainnya dan melakukan seleksi kompetitif untuk memilih anggota populasi sebagai induk agar ditahap berikutnya siap untuk melakukan kawin silang. Tahap keempat adalah melakukan kawin silang antar induk yang terpilih lalu menyeleksi *child* terbaik yang dihasilkan dari kawin silang tersebut dan melakukan mutasi terhadap *child* terbaik tersebut. Tahap kelima menggabungkan semua populasi dengan semua *child*. Tahap keenam dengan menggunakan fungsi *fitness* hitung nilai *makespan* dan urutkan secara *ascending* berdasarkan *makespan*. Diagram alir proses GA ditunjukkan oleh Gambar 2.5.

2.2.1 Membangkitkan Kromosom

Pada permasalahan penjadwalan NWFS, urutan dari beberapa pekerjaan yang diproses oleh serangkaian mesin diterjemahkan sebagai satuan kromosom-kromosom. GA berfungsi untuk mencari urutan-urutan kerja yang memenuhi ketentuan yang dicari. Kumpulan dari beberapa kromosom tersebut disebut sebagai populasi. Kromosom dibangkitkan dengan acak secara permutasi sesuai data uji yang digunakan. Jumlah kromosom disesuaikan dari banyaknya jumlah populasi yang ditentukan sebelumnya dan panjang kromosom disesuaikan dengan banyaknya jumlah pekerjaan yang terdapat pada data uji.

Contoh bentuk kromosom dari data uji coba pada Tabel 2.1 yang sudah dibangkitkan ditunjukkan oleh Tabel 2.3.

Tabel 2.3 Contoh Bentuk Kromosom

Nomor Kromosom	Urutan Kromosom
1	1 – 2 – 3
2	1 – 3 – 2
3	2 – 1 – 3
4	2 – 3 – 1
5	3 – 1 – 2
6	3 – 2 – 1

2.2.2 Kawin Silang atau *Crossover*

Kawin silang atau biasa disebut dengan *crossover* adalah proses pengawinan silang antara dua *parent* untuk mendapatkan kombinasi yang terbaik dari satu kromosom dengan kromosom lainnya. Terdapat macam-macam teknik kawin silang pada Algoritma Genetika, seperti *One-Point Crossover*, *K-Point Crossover*, *Uniform Crossover*, *Discrete Crossover* dan lain-lain [7].

Teknik kawin silang yang digunakan pada permasalahan NWFS adalah *K-Point crossover*. Banyaknya operator yang digunakan untuk melakukan kawin silang adalah dua operator. Posisi dua operator tersebut dibangkitkan secara acak seperti yang ditunjukkan oleh Persamaan 2.14. Dua posisi tersebut memiliki ketentuan, yaitu harus bernilai positif dan nilainya tidak boleh melebihi panjang kromosom. Ketentuan tersebut ditunjukkan oleh Persamaan 2.15. Proses kawin silang itu sendiri ditunjukkan oleh Persamaan 2.16.

$$Q_1 = [c_1, c_2] \quad (2.14)$$

$$0 \leq c_1 < c_2 \leq 1 \quad (2.15)$$

$$A_i^k = Q_1(X_i^k \otimes X_i^{k+1}) \quad (2.16)$$

Kombinasi linier parameter pada kawin silang adalah probabilitas kawin silang. Kawin silang yang terjadi pada suatu populasi bergantung pada parameter tersebut. Apabila parameter tersebut bernilai kecil, maka kawin silang yang terjadi pada populasi tersebut menjadi sedikit. Sebaliknya apabila parameter tersebut bernilai besar, maka kromosom yang mengalami kawin silang pada populasi tersebut menjadi banyak. Sebagai contoh apabila nilai parameter tersebut 0.61, maka terjadi kemungkinan 61% kawin silang pada populasi tersebut. Parameter kawin silang dianjurkan untuk memiliki nilai yang besar atau mendekati satu karena nilai yang besar membuat probabilitas kawin silang menjadi lebih banyak. Sehingga semakin banyak terjadinya kawin silang, maka semakin banyaknya rangkaian kromosom yang berbeda. Anjuran tersebut dikemukakan oleh Wen-Yang Ling, Wen-Yuan Lee, dan Tzung-Pei Hong [8]. Parameter tersebut mempunyai ketentuan seperti yang ditunjukkan oleh Persamaan 2.17. $child_1 < child_2$

$$0 \leq cr \leq 1 \quad (2.17)$$

Kawin silang dapat terjadi apabila memenuhi kondisi $r_1 \leq cr$. Apabila kondisi tersebut tidak dapat dipenuhi, maka kromosom tidak perlu melakukan kawin silang. Agar memudahkan untuk memahami proses pada kawin silang, maka dijelaskan proses kawin silang secara keseluruhan oleh Tabel 2.4. Hasil dari proses kawin silang tersebut adalah dua *child*. Dengan menggunakan fungsi *fitness*, dua *child* tersebut dihitung nilai *makespan*-nya. Penyeleksian dilakukan terhadap *child* yang memiliki *makespan* lebih kecil. *Child* yang memiliki *makespan* lebih kecil digabungkan kedalam populasi awal.

Tabel 2.4 Proses Kawin Silang

Langkah	Proses	Contoh
1	Tentukan $parent_1$ dan $parent_2$ dari populasi	$parent_1 = 5-7-4-1-2-3-6$ $parent_2 = 6-3-7-1-5-4-2$
2	Tentukan titik crossover secara acak	$C_1 = 3$ $C_2 = 5$
3	Salin posisi $C_1, C_1 + 1, \dots, C_2$ milik $parent_1$ dan $parent_2$ kedalam variabel $child_1$ dan $child_2$.	$child_1 = _ _ 4 \ 1 \ 2 \ _$ $_$ $child_2 = _ _ 7 \ 1 \ 5 \ _$ $_$
4	Masukkan elemen-elemen yang terdapat pada $parent_1$ tetapi tidak ada pada $child_2$ kedalam variabel $child_2$ searah jarum jam.	$child_2 = 6-4-7-1-5-2-3$
5	Masukkan elemen-elemen yang terdapat pada $parent_2$ tetapi tidak ada pada $child_1$ kedalam variabel $child_1$ searah jarum jam.	$child_1 = 7-5-4-1-2-6-3$

2.2.3 Mutasi

Mutasi pada GA memiliki tujuan untuk mendapatkan bentuk kromosom yang tidak didapat pada proses kawin silang. Mutasi dilakukan setelah proses kawin silang selesai. Proses mutasi terjadi dengan cara menukar dua posisi pekerjaan pada kromosom seperti yang ditunjukkan pada Persamaan 2.18. Persamaan tersebut menjelaskan mutasi yang terjadi pada urutan

kerja terhadap setiap kromosom. Persamaan 2.19 menunjukkan dua posisi yang bertukar. Dua posisi tersebut dibangkitkan secara acak dengan ketentuan harus bernilai positif dan tidak melebihi panjang kromosom. Ketentuan tersebut ditunjukkan oleh Persamaan 2.20.

$$B_t^k = Q_2(A_t^{k-1}) \quad (2.18)$$

$$Q_2 = [pm_1, pm_2] \quad (2.19)$$

$$1 \leq pm_1 < pm_2 \leq n \quad (2.20)$$

Mutasi memiliki parameter untuk membatasi jumlah kromosom yang dimutasi. Parameter tersebut memiliki ketentuan seperti berikut, yaitu harus bilangan bernilai positif dan bilangan yang berada diantara nol hingga satu seperti yang ditunjukkan oleh Persamaan 2.21.

$$0 \leq m \leq 1 \quad (2.21)$$

Apabila parameter tersebut bernilai kecil, maka hanya sedikit mutasi yang terjadi pada populasi tersebut. Sebaliknya apabila parameter tersebut bernilai besar, maka kromosom yang mengalami mutasi pada populasi tersebut menjadi banyak. Sebagai contoh apabila nilai parameter mutasi 0,11, maka terjadi kemungkinan 11% mutasi pada populasi tersebut. Parameter mutasi dianjurkan untuk memiliki nilai yang kecil karena apabila banyak kromosom yang terkena mutasi menyebabkan hasil bentuk kromosom yang sama. Anjuran tersebut dikemukakan oleh Wen-Yang Ling, Wen-Yuan Lee, dan Tzung-Pei Hong [8].

Tabel 2.5 Proses Mutasi

Kondisi Awal	6	4	7	1	5	2	3
Kondisi Akhir	6	2	7	1	5	4	3

Mutasi dapat terjadi apabila memenuhi kondisi $r_1 \leq m$. Apabila kondisi tersebut tidak dapat dipenuhi, maka kromosom tidak perlu dimutasi. Ilustrasi mutasi dapat dilihat pada Tabel 2.5.

2.3 Pendekatan *Tabu Search* Pada Permasalahan Penjadwalan *No-Wait Flow Shop*

Metode *Tabu Search* (TS) diperkenalkan pada tahun 1986 oleh Fred W. Glover [2]. TS adalah metode optimasi metaheuristik yang berbasis pada *local search*. Konsep dasar dari TS adalah proses pencegahan pencarian dari *local search* agar tidak melakukan pengulangan pencarian pada ruang solusi yang sudah pernah ditelusuri dengan memanfaatkan struktur memori. Struktur memori tersebut mencatat jejak proses pencarian yang telah dilakukan. Pada penelitian ini, TS digunakan untuk menghindarkan metode GA dari kemungkinan terjebak pada kondisi *local optima* [2]. Metode TS dijalankan setelah pembangkitan kromosom dan sebelum proses kawin silang pada GA.

Proses TS mengambil beberapa kromosom dari keseluruhan kromosom yang terdapat pada populasi. Proses TS adalah memutasi kromosom-kromosom yang terpilih dengan tujuan agar mendapatkan nilai terbaik dari kromosom yang terdekat dengan kromosom-kromosom tersebut. Setelah proses mutasi dilakukan, proses selanjutnya adalah menghitung nilai *makespan* dari kromosom yang baru dibentuk. Nilai *makespan* tersebut dibandingkan dengan nilai *makespan* dari kromosom saat sebelum dimutasi. Apabila *makespan* dari kromosom terbaru memiliki nilai yang lebih kecil, maka bentuk kromosom baru tersebut diterima dan disimpan sebagai bentuk kromosom baru. Apabila *makespan* kromosom tersebut lebih besar, maka dilakukan perhitungan kemungkinan penerimaan kromosom baru tersebut. Perhitungan tersebut ditunjukkan oleh Persamaan 2.22. Kromosom tersebut dapat diterima menjadi kromosom baru apabila memenuhi kondisi seperti yang ditunjukkan oleh Persamaan 2.23. Proses mutasi

tersebut berhenti ketika telah mencapai nilai batas iterasi TS. Penurunan nilai iterasi TS ditunjukkan oleh Persamaan 2.24.

$$TS = \exp\left(\frac{-(makespan\ baru - makespan\ lama)}{I(k)}\right) \quad (2.22)$$

$$TS \geq Y \quad (2.23)$$

$$I(k) = I_0 \times \alpha^k \quad (2.24)$$

Ketika proses TS sudah selesai dan mencapai batas iterasi, maka kromosom-kromosom baru yang sudah disimpan sebelumnya ditambahkan kedalam populasi awal kromosom. Hal ini menyebabkan jumlah kromosom pada populasi bertambah banyak. Penggunaan proses TS ini menghindari *local optima* pada metode GA.

BAB III

PERANCANGAN PERANGKAT LUNAK

Bab ini membahas mengenai perancangan dan pembuatan sistem perangkat lunak. Sistem perangkat lunak yang dibuat pada penelitian ini adalah implementasi penyelesaian *No-Wait Flow Shop* dengan menggunakan metode Algoritma Genetika dan *Tabu Search*. Perancangan perangkat lunak pada bab ini dibagi menjadi empat bagian, yaitu penjelasan variabel yang digunakan pada perancangan penyelesaian NWFS dengan menggunakan metode GA-TS, penjelasan perancangan metode GA-TS untuk mendapatkan urutan pekerjaan terbaik, penjelasan perancangan metode GA-TS untuk mendapatkan *makespan* terbaik, dan penjelasan perancangan metode GA-TS untuk permasalahan NWFS secara umum.

3.1 Variabel yang Digunakan Pada Perancangan Penyelesaian *No-Wait Flow Shop* Menggunakan Metode GA-TS

Subbab ini menjelaskan mengenai variabel-variabel yang digunakan dalam *pseudocode* pada penyelesaian NWFS dengan menggunakan metode GA-TS. Variabel-variabel tersebut meliputi nama variabel, tipe variabel, dan penjelasan variabel sebagaimana ditunjukkan oleh Tabel 3.1, Tabel 3.2, Tabel 3.3, Tabel 3.4, Tabel 3.5, Tabel 3.6, dan Tabel 3.7.

Tabel 3.1 Daftar Variabel yang Digunakan pada Perancangan Penyelesaian NWFS Menggunakan Metode GA-TS (Bagian 1)

No.	Variabel	Tipe	Penjelasan
1.	<i>data</i>	<i>double</i>	Data uji yang dimasukkan kedalam sistem

**Tabel 3.2 Daftar Variabel yang Digunakan pada
Perancangan Penyelesaian NWFS Menggunakan Metode
GA-TS (Bagian 2)**

No.	Variabel	Tipe	Penjelasan
2.	<i>mesin</i>	<i>int</i>	Jumlah mesin yang terdapat pada <i>dataset</i>
3.	<i>job</i>	<i>int</i>	Jumlah pekerjaan yang terdapat pada <i>dataset</i>
4.	<i>Populasi</i>	<i>int</i>	Jumlah populasi yang dibangkitkan dari <i>dataset</i>
5.	<i>m</i>	<i>double</i>	Parameter untuk menentukan kromosom yang dimutasi
6.	<i>cr</i>	<i>double</i>	Parameter untuk menentukan kromosom yang dikawin silang
7.	<i>parTS</i>	<i>double</i>	Parameter pengambilan kromosom sampel untuk metode TS
8.	<i>iterasiTS</i>	<i>double</i>	Nilai awal iterasi pada TS
9.	<i>batasIterasiTS</i>	<i>double</i>	Batas akhir iterasi pada TS
10.	<i>alphaTS</i>	<i>double</i>	Koefisien <i>alpha</i> untuk penurunan iterasi
11.	<i>kromosom</i>	<i>int</i>	Matriks yang berisikan kromosom yang berasal dari pembangkitan acak
12.	<i>timeKromosom</i>	<i>int</i>	Matriks yang berisikan <i>makespan</i> yang berasal dari pembangkitan acak

**Tabel 3.3 Daftar Variabel yang Digunakan pada
Perancangan Penyelesaian NWFS Menggunakan Metode
GA-TS (Bagian 3)**

No.	Variabel	Tipe	Penjelasan
13.	<i>waktuNWFS</i>	<i>int</i>	Matriks yang berisikan <i>makespan</i> yang telah diurutkan
14.	<i>parent</i>	<i>int</i>	Matriks yang berisikan kromosom yang telah diurutkan berdasarkan <i>makespan</i>
15.	<i>urutan</i>	<i>int</i>	Nilai untuk mengurutkan kromosom
16.	<i>kromosomTS</i>	<i>int</i>	Matriks yang berisikan urutan kromosom sampel pada metode TS
17.	<i>waktuTS</i>	<i>int</i>	Matriks yang berisikan <i>makespan</i> dari kromosom sampel pada metode TS
18.	<i>middleTS</i>	<i>int</i>	Parameter pengambilan sampel TS bagian tengah
19.	<i>downTS</i>	<i>int</i>	Parameter pengambilan sampel TS bagian bawah
20.	<i>jumlahMutasi</i>	<i>int</i>	Banyaknya mutasi yang dilakukan
21.	<i>tempKromosomTS</i>	<i>int</i>	Kromosom sementara untuk dimutasi pada metode TS
22.	<i>waktuMutasi</i>	<i>double</i>	<i>Makespan</i> kromosom hasil mutasi

**Tabel 3.4 Daftar Variabel yang Digunakan pada
Perancangan Penyelesaian NWFS Menggunakan Metode
GA-TS (Bagian 4)**

No.	Variabel	Tipe	Penjelasan
23.	<i>TS</i>	<i>double</i>	Kemungkinan penerimaan kromosom setelah mutasi pada metode TS
24.	<i>y</i>	<i>double</i>	Nilai yang dihasilkan acak dengan rentang 0 hingga 1
25.	<i>availCR</i>	<i>int</i>	Matriks yang berisikan indeks kromosom yang siap untuk kawin silang
26.	<i>iterCR</i>	<i>int</i>	Banyaknya kawin silang yang dilakukan
27.	<i>child1</i>	<i>int</i>	kromosom hasil dari kawin silang
28.	<i>child2</i>	<i>int</i>	kromosom hasil dari kawin silang
29.	<i>waktuChild1</i>	<i>int</i>	<i>Makespan</i> yang didapat dari kromosom hasil kawin silang
30.	<i>waktuChild2</i>	<i>int</i>	<i>Makespan</i> yang didapat dari kromosom hasil kawin silang
31.	<i>availM</i>	<i>int</i>	Matriks yang berisikan indeks kromosom yang siap untuk dimutasi
32.	<i>iterM</i>	<i>int</i>	Banyaknya mutasi yang dilakukan

**Tabel 3.5 Daftar Variabel yang Digunakan pada
Perancangan Penyelesaian NWFS Menggunakan Metode
GA-TS (Bagian 5)**

No.	Variabel	Tipe	Penjelasan
33.	$r1$	<i>double</i>	Nilai yang dihasilkan acak dengan rentang 0 hingga 1
34.	$r2$	<i>double</i>	Nilai yang dihasilkan acak dengan rentang 0 hingga 1
35.	$posisi$	<i>int</i>	Matriks yang berisikan nilai acak dengan rentang 1 hingga jumlah pekerjaan
36.	$p1$	<i>int</i>	Operator posisi pertama yang digunakan untuk kawin silang dan mutasi
37.	$p2$	<i>int</i>	Operator posisi kedua yang digunakan untuk kawin silang dan mutasi
38.	$mutasiTemp$	<i>int</i>	Matriks yang berisikan kromosom sementara untuk dimutasi
39.	$waktuKromosomFinal$	<i>int</i>	Matriks yang berisikan nilai <i>makespan</i> setelah penggabungan <i>parent</i> dan <i>child</i>
40.	$waktu$	<i>int</i>	<i>Makespan</i> sementara saat penghitungan kromosom
41.	$bestWaktu$	<i>int</i>	<i>Makespan</i> terbaik

**Tabel 3.6 Daftar Variabel yang Digunakan pada
Perancangan Penyelesaian NWFS Menggunakan Metode
GA-TS (Bagian 6)**

No.	Variabel	Tipe	Penjelasan
42.	<i>bestKromosom</i>	<i>int</i>	Urutan kerja terbaik
43.	<i>delay</i>	<i>int</i>	Matriks yang berisikan waktu tunggu antar pekerjaan
44.	<i>sumDelay</i>	<i>int</i>	Jumlah nilai waktu tunggu pada suatu kromosom
45.	<i>sumJobRow</i>	<i>int</i>	Jumlah nilai dari setiap pekerjaan dari mesin pertama hingga terakhir
46.	<i>compare</i>	<i>int</i>	Nilai perbandingan dari Persamaan 2.9
47.	<i>length</i>	<i>int</i>	Panjang dari kromosom
48.	<i>counter1</i>	<i>int</i>	Inisiasi nilai untuk memasukkan posisi urutan kerja yang sedang dikawin silang
49.	<i>counter2</i>	<i>int</i>	Inisiasi nilai untuk memasukkan posisi urutan kerja yang sedang dikawin silang
50.	<i>tempc</i>	<i>int</i>	Kromosom sementara dari hasil kawin silang
51.	<i>tempd</i>	<i>int</i>	Kromosom sementara dari hasil kawin silang
52.	<i>crossover</i>	<i>double</i>	Fungsi kawin silang untuk kromosom

**Tabel 3.7 Daftar Variabel yang Digunakan pada
Perancangan Penyelesaian NWFS Menggunakan Metode
GA-TS (Bagian 7)**

No.	Variabel	Tipe	Penjelasan
53.	<i>fitness</i>	<i>double</i>	Fungsi untuk mendapatkan <i>makespan</i>
54.	<i>slice</i>	<i>double</i>	Fungsi untuk mencari pekerjaan yang sama pada dua kromosom yang dikawin silang

3.2 Perancangan Metode GA-TS Untuk Mendapatkan Urutan Pekerjaan Terbaik

Pada subbab ini dijelaskan mengenai perancangan metode GA-TS untuk mendapatkan urutan pekerjaan terbaik. Tujuan dari metode GA-TS adalah untuk mendapatkan *makespan* terkecil dari urutan pekerjaan yang terbaik yang di mana urutan pekerjaan terbaik didapatkan dengan beberapa langkah. Langkah-langkah untuk mendapatkan urutan terbaik adalah membangkitkan kromosom, pengambilan sampel kromosom untuk metode TS, memproses metode TS pada kromosom, melakukan kawin silang pada kromosom, memutasi kromosom, dan mengurutkan kromosom berdasarkan *makespan* terbaik.

3.2.1 Perancangan Pembangkitan Kromosom

Pembangkitan kromosom adalah langkah pertama untuk mendapatkan urutan pekerjaan yang terbaik. Pembangkitan kromosom dilakukan dengan menggunakan fungsi *random permutation*. Kromosom yang dibangkitkan memiliki urutan-urutan yang berbeda pada setiap kromosom. Jumlah kromosom yang dibangkitkan adalah sebanyak populasi yang sudah ditetapkan. Panjang dari kromosom-kromosom tersebut sama

dengan banyaknya pekerjaan yang terdapat pada data uji coba. Perancangan pembangkitan kromosom pada *pseudocode* ditunjukkan oleh Gambar 3.1.

Input	Jumlah Pekerjaan (job)
Output	Matriks yang berisikan kromosom yang berasal dari pembangkitan acak
1	for i = 1 : populasi
2	temp = randperm(job)
3	for j = 1 : job
4	kromosom (i, j) = temp (1, j)
5	endfor
6	endfor

Gambar 3.1 Pseudocode Pembangkitan Kromosom Awal

3.2.2 Perancangan Pengambilan Sampel Kromosom Untuk Metode TS

Pengambilan sampel kromosom bertujuan untuk mendapatkan seluruh kromosom tetapi tidak perlu semua kromosom. Hal itu disebabkan karena sudah dapat diwakilkan dari tiga kelompok sampel yang diambil. Pembagian tiga kelompok tersebut berdasarkan nilai *makespan*. Tiga kelompok tersebut, yaitu kelompok pertama adalah kelompok yang memiliki nilai *makespan* kecil, kelompok kedua adalah kelompok yang memiliki nilai *makespan* sedang, dan kelompok ketiga adalah kelompok yang memiliki nilai *makespan* besar. Perancangan pengambilan sampel kromosom tersebut terdapat pada *pseudocode* yang ditunjukkan oleh Gambar 3.2 dan Gambar 3.3.

Input	Urutan kerja (parent), <i>makespan</i> (waktuNWFS), parameter TS (partS)
Output	Urutan kerja untuk mutasi TS (kromosomTS)

Gambar 3.2 Pseudocode Pengambilan Sampel Kromosom (Bagian 1)

```

1  for i = 1 : partS
2      kromosomTS(i,:) = parent(i,:)
3      waktuTS(i) = waktuNWFS(i,1)
4  endfor
5  middleTS = round(populasi/2) -
    floor(partS/2)
6  for i = partS+1 : partS*2
7      kromosomTS(i,:) = parent(middleTS,:)
8      waktuTS(i) = waktuNWFS(middleTS,1)
9      middleTS = middleTS + 1;
10 endfor
11 downTS = populasi+1 - partS
12 for i = partS*2 + 1 : partS*3
13     kromosomTS(i,:) = parent(downTS,:)
14     waktuTS(i) = waktuNWFS(downTS,1)
15     downTS = downTS + 1;
16 endfor

```

**Gambar 3.3 Pseudocode Pengambilan Sampel Kromosom
(Bagian 2)**

Seleksi kromosom untuk sampel dari kelompok kecil diterapkan pada baris ke 1 hingga 4, kelompok sedang diterapkan dari baris ke 5 hingga 10, dan kelompok besar diterapkan dari baris ke 11 hingga 16. Dari pembagian kelompok tersebut diharapkan sudah mewakili seluruh kromosom secara garis besar sebelum dimutasi pada proses TS.

3.2.3 Perancangan Proses Metode TS

Metode TS dilakukan agar tidak terjebak pada kondisi *local optima* GA. Mutasi dilakukan sebanyak kromosom sampel yang ada. Proses mutasi dilakukan pada metode TS yang mana nantinya membandingkan *makespan* dari hasil sebelum dan sesudah TS. Apabila nilai *makespan* dari hasil mutasi TS tidak lebih kecil, maka terdapat ketentuan untuk dapat tetap menjadikan hasil mutasi TS tersebut menjadi salah satu kromosom baru. Ketentuan tersebut sudah ditunjukkan pada Persamaan 2.22 yang

diterapkan pada baris ke 15. Nilai yang dihasilkan dari Persamaan 2.22 dibandingkan dengan suatu nilai acak, seperti yang ditunjukkan oleh Persamaan 2.23 dan diterapkan pada baris ke-17. Iterasi pada metode TS terus dilakukan hingga nilai iterasi kurang dari nilai batas iterasi. Perhitungan penurunan nilai iterasi ditunjukkan oleh Persamaan 2.24 dan diterapkan pada baris ke-22. Hasil dari kromosom baru yang terbentuk dimasukkan kedalam populasi kromosom. *Pseudocode* perancangan metode TS ditunjukkan oleh Gambar 3.4 dan Gambar 3.5.

Input	Urutan kerja (kromosomTS)
Output	Urutan kerja hasil mutasi TS (kromosomTS), <i>makespan</i> hasil mutasi TS (waktuMutasi)
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	<pre> while (iterasiTS > batasIterasiTS) jumlahMutasi = size(kromosomTS, 1) for i=1:jumlahMutasi posisi = randperm(job,2) p1 = min (posisi) p2 = max (posisi) tempKromosomTS(i,:) = kromosomTS(i,:) mutasiTemp = tempKromosomTS(i,p1) tempKromosomTS(i,p1) = tempKromosomTS(i,p2); tempKromosomTS(i,p2) = mutasiTemp [waktuMutasi(i)] = fitness2(data,tempKromosomTS(i,:),mesin,job if waktuMutasi(1, i) <= waktuTS(1,i) kromosomTS(i,:) = tempKromosomTS(i,:) else TS = exp((- (waktuMutasi(1,i) - waktuTS(1,i))) / iterasiTS) </pre>

Gambar 3.4 *Pseudocode* Proses Metode TS (Bagian 1)

16	y = rand
17	if TS >= y
18	kromosomTS(i,:) =
	tempKromosomTS(i,:)
19	endif
20	endif
21	endfor
22	iterasiTS = iterasiTS * alphaTS ^ ii
23	endwhile

Gambar 3.5 Pseudocode Proses Metode TS (Bagian 2)

3.2.4 Perancangan Proses GA

Proses pada GA dibagi menjadi dua subbab, yaitu proses kawin silang dan proses mutasi. Metode GA dibutuhkan untuk mencari urutan-urutan kerja yang belum dimiliki pada kromosom sebelumnya, sehingga kromosom harus di kawin silangkan dengan kromosom tetangga dan dimutasi. Proses kawin silang pada GA dijelaskan pada subbab 3.2.4.1 dan proses mutasi pada GA dijelaskan pada subbab 3.2.4.2.

3.2.4.1 Perancangan Proses Kawin Silang

Proses kawin silang pada permasalahan NWFS melakukan kawin silang pada kromosom-kromosom tetangganya. Hasil kawin silang diseleksi berdasarkan nilai *makespan* yang lebih kecil seperti yang terdapat pada baris ke-40 dan hasil tersebut disimpan kedalam variabel hasil dari kawin silang, seperti yang terdapat pada baris ke-41 dan 43. Sebelum terjadi proses kawin silang, kromosom diseleksi terlebih dahulu berdasarkan parameter kawin silang yang memiliki ketentuan seperti pada Persamaan 2.17 yang diterapkan pada baris ke-1 hingga 5. Proses kawin silang berdasarkan pada Tabel 2.4 dan Tabel 2.5. Proses kawin silang diterapkan pada baris ke-7 hingga 37. *Pseudocode* dari perancangan proses kawin silang ditunjukkan oleh Gambar 3.6 dan Gambar 3.7

Input	Urutan kerja sebagai kromosom pertama (parent1), urutan kerja sebagai kromosom kedua (parent2), parameter kawin silang (cr)
Output	Kromosom dari hasil kawin silang
1	for i = 1 : populasi
2	r1(i) = rand
3	endfor
4	availCR = find (r1 < cr)
5	iterCR = size(availCR,2)
6	for i = 1 : iterCR
7	counter1 = 1
8	counter2= 1
9	length = size(parent1, 2)
10	posisi = randperm (length-1, 2)
11	p1 = min (posisi)
12	if p1 == 1
13	p1 = p1 + 1
14	endif
15	p2 = max (posisi)
16	if p2 == p1
17	p2 = p2 + 1
18	endif
19	child1 = zeros (1, length)
20	child2 = zeros (1, length)
21	for j = p1 : p2
22	child1(j) = parent1(j)
23	child2(j) = parent2(j)
24	endfor
25	[tempc tempd] = slice(parent1, child1, parent2, child2, p1, p2)
26	for j = p2 + 1 : length
27	child1(j) = tempc(counter1)
28	child2(j) = tempd(counter2)
29	counter1 = counter1 + 1
30	counter2 = counter2 + 1

**Gambar 3.6 Pseudocode Proses Kawin Silang pada GA
(Bagian 1)**

```

31      endfor
32      for j = 1 : p1 - 1
33          child1(j) = tempc(counter1)
34          child2(j) = tempd(counter2)
35          counter1 = counter1 + 1
36          counter2 = counter2 + 1
37      endfor
38      [waktuChild1] =
fitness2(data,child1,mesin,job)
39      [waktuChild2] =
fitness2(data,child2,mesin,job)
40      if waktuChild1 < waktuChild2
41          child(i,:) = child1
42      else
43          child(i,:) = child2
44      endif
45  endfor

```

**Gambar 3.7 Pseudocode Proses Kawin Silang pada GA
(Bagian 2)**

Penggunaan operator untuk melakukan kawin silang berdasarkan Persamaan 2.14 dan Persamaan 2.15 yang diterapkan pada baris ke-10 hingga 18. Penyalinan urutan pekerjaan yang berada diantara dua operator dari $parent_1$ kedalam $child_1$ dan $parent_2$ ke $child_2$ diterapkan pada baris ke-19 hingga 24. Pencarian dan penyeleksian pekerjaan yang sama diterapkan pada baris ke-25. Proses pembentukan *child* yang baru diterapkan pada baris ke-26 hingga 36. Proses seleksi *child* yang memiliki *makespan* terkecil diterapkan pada baris ke-38 hingga 44.

3.2.4.2 Perancangan Proses Mutasi Metode GA

Proses mutasi bertujuan untuk mendapatkan urutan pekerjaan baru yang sebelumnya tidak didapat pada saat melakukan kawin silang. Diharapkan dengan dilakukannya mutasi didapatkan urutan pekerjaan baru dan memiliki *makespan* yang lebih baik. Proses mutasi dilakukan berdasarkan Persamaan 2.18,

Persamaan 2.19, dan Persamaan 2.20. Sebelum terjadi proses mutasi, kromosom diseleksi terlebih dahulu berdasarkan parameter mutasi yang memiliki ketentuan seperti pada Persamaan 2.21 yang diterapkan pada baris ke-1 hingga 5. *Pseudocode* perancangan proses mutasi pada GA ditunjukkan oleh Gambar 3.8.

Input	Urutan kerja (kromosom), banyaknya pekerjaan (job), parameter mutasi (m)
Output	Urutan kerja hasil mutasi (child)
1	for i=1 : size(child,1)
2	r2(i) = rand
3	endfor
4	availM = find (r2 < m)
5	iterM = size(availM,2)-1
6	for i = 1 : iterM
7	posisi = randperm(job,2)
8	p1 = min (posisi)
9	p2 = max (posisi)
10	mutasiTemp = child (availM(i),p1)
11	child (availM(i),p1) = child
12	child (availM(i),p2) = mutasiTemp
13	endfor

Gambar 3.8 Pseudocode Proses Mutasi pada GA (Bagian 1)

Pembangkitan operator mutasi secara acak diterapkan pada baris ke-7 hingga 9. Proses penukaran posisi pada kromosom ditunjukkan pada baris ke-10 hingga 12.

3.2.5 Perancangan Pengurutan Kromosom Berdasarkan *Makespan* Terbaik

Pengurutan kromosom berdasarkan *makespan* terbaik memiliki tujuan untuk mendapatkan urutan kerja dan *makespan* terbaik. Nilai *makespan* yang didapat dari kromosom-kromosom beraneka ragam. Agar mendapatkan *makespan* dan kromosom terbaik, maka diurutkan kromosom tersebut secara *ascending*. Dengan demikian, *makespan* terbaik dan kromosom terbaik berada

di posisi paling atas. Pengurutan nilai *makespan* diterapkan pada baris pertama sedangkan pengurutan kromosom yang memiliki *makespan* terbaik diterapkan pada baris ke-2 hingga 4. *Pseudocode* untuk perancangan pengurutan kromosom berdasarkan *makespan* terbaik secara *ascending* ditunjukkan oleh Gambar 3.9.

Input	Urutan kerja (kromosom), <i>makespan</i> (timeKromosom)
Output	Kromosom yang sudah diurutkan (parent), <i>makespan</i> yang sudah diurutkan (waktuNWFS)
1	[waktuNWFS urutan] = sort (timeKromosom)
2	for i = 1 : populasi
3	parent(i,:) = kromosom(urutan(i, 1),:)
4	endfor
5	[waktuNWFS urutan] = sort (waktuNWFS)

Gambar 3.9 Pengurutan Kromosom Berdasarkan *Makespan*

3.3 Perancangan Metode GA-TS Untuk Mendapatkan Nilai *Makespan*

Subbab ini menjelaskan mengenai perancangan metode GA-TS untuk mendapatkan nilai *makespan*. Perancangan ini juga untuk menyelesaikan fungsi batasan pada permasalahan NWFS. Batasan-batasan yang harus dilakukan adalah setiap pekerjaan harus dikerjakan oleh satu atau lebih mesin, setiap pekerjaan harus diproses hingga selesai oleh satu mesin sebelum dapat berpindah ke mesin berikutnya sesuai urutannya, dan setiap mesin harus menyelesaikan pekerjaan yang sedang dikerjakan terlebih dahulu sebelum dapat memproses pekerjaan lainnya. Batasan-batasan tersebut dijelaskan pada Persamaan 2.5, Persamaan 2.6, dan Persamaan 2.7 secara berurutan. Perhitungan mendapatkan *makespan* juga harus memenuhi kondisi-kondisi seperti yang ditunjukkan pada Persamaan 2.10, Persamaan 2.11, dan Persamaan 2.12. Perhitungan untuk mendapatkan nilai *makespan* dijelaskan pada subbab setelah ini.

3.3.1 Perancangan Perhitungan Nilai *Makespan*

Pada subbab ini dijelaskan mengenai perhitungan untuk mendapatkan nilai *makespan*. Perhitungan nilai *makespan* terjadi beberapa kali pada metode GA-TS untuk menyelesaikan permasalahan NWFS. Pada baris ke-1 hingga 12, baris tersebut menjelaskan mengenai perhitungan untuk mendapatkan waktu tunggu seperti yang terdapat pada Persamaan 2.9. *Pseudocode* perancangan perhitungan nilai *makespan* ditunjukkan oleh Gambar 3.10 dan Gambar 3.11.

Input	Urutan kerja (parent)
Output	<i>Makespan</i> (waktu)
1	for i = 1 : job
2	for k = 1 : job
3	if i == k
4	delay(i,k) = 0
5	else
6	temp = data(i,2) - data(k,1)
7	temp2 = sum(data(i,2:mesin),2)
	- sum(data(k,1:mesin-1),2)
8	compare = [temp2 temp 0]
9	delay(i,k) = max(compare)
10	endif
11	endfor
12	endfor
13	for j=1 : job
14	for i=1 : mesin
15	dataKromosom (j,i) =
	data(kromosom(1, j), i);
16	endfor
17	endfor
18	for j = 1 : job
19	if j==1

Gambar 3.10 Pseudocode Perhitungan Nilai *Makespan* (Bagian 1)

```

20         for i = 1 : mesin
21             if i == 1
22                 waktu(j, i) = dataKromosom
23                 (j,i);
24             else
25                 waktu (j, i) = waktu(j, i-
26                 1) + dataKromosom (j,i);
27             endif
28         endfor
29     for j=2:job
30         waktu (j, 1) = waktu(j-1, 1) +
31         delay(kromosom(1,j-1),kromosom(1,j)) +
32         dataKromosom(j,1);
33     endfor
34     for j=2:job
35         for i = 2 : mesin
36             if waktu(j, i-1) > waktu (j-1, i)
37                 waktu(j, i) = waktu(j, i-1) +
38                 dataKromosom (j,i);
39             else
40                 waktu(j, i) = waktu(j-1, i) +
41                 dataKromosom (j,i);
42             endif
43         endfor
44     endfor
45     waktu = waktu(j,i);

```

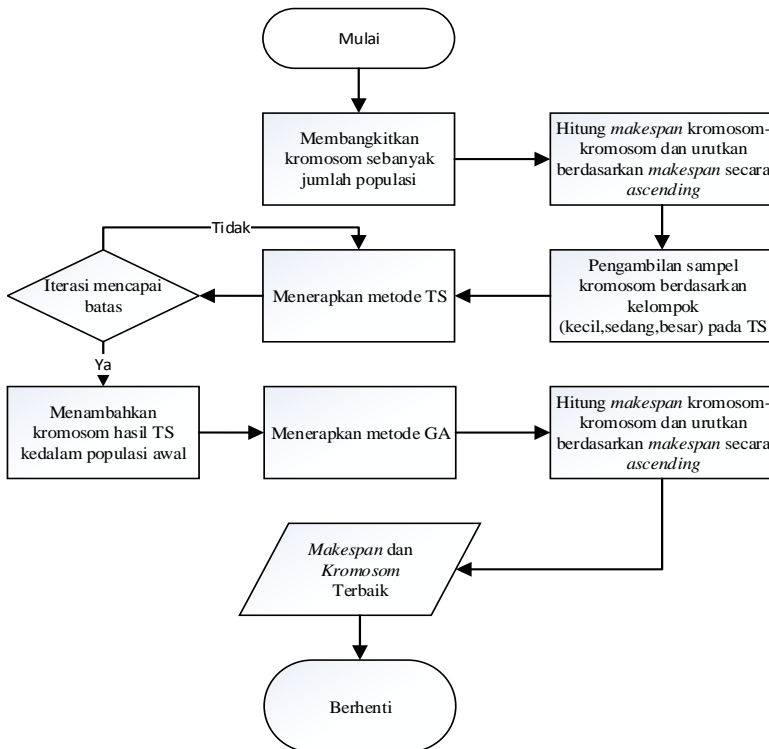
Gambar 3.11 Pseudocode Perhitungan Nilai Makespan (Bagian 2)

Pemindahan data agar sesuai dengan urutan kromosom ditunjukkan pada baris ke-13 hingga 17. Penjumlahan setiap mesin pada pekerjaan pertama ditunjukkan pada baris ke-18 hingga 28 dan penjumlahan waktu setiap pekerjaan dimulai dari mesin kedua ditunjukkan pada baris ke-19 hingga 23. Penjumlahan waktu tunggu dengan waktu selesai mesin pertama terdapat pada baris-29 hingga 31. Perhitungan waktu proses sesuai dengan *directed*

graph dan sesuai dengan Persamaan 2.10, Persamaan 2.11, dan Persamaan 2.12 ditunjukkan pada baris ke-32 hingga 40. Nilai *makespan* didapatkan pada baris ke-41.

3.4 Perancangan Metode GA-TS Untuk Permasalahan *No-Wait Flow Shop* Secara Umum

Subbab ini menjelaskan mengenai perancangan metode GA-TS untuk permasalahan NWFS secara umum.



Gambar 3.12 Diagram alir proses Penyelesaian NWFS dengan Menggunakan Metode GA-TS Secara Umum

Penjelasan tersebut dijelaskan dengan diagram alir proses keseluruhan penyelesaian NWFS dengan menggunakan metode GA-TS dan dengan *pseudocode* umum metode GA-TS pada permasalahan NWFS. Diagram alir ditunjukkan pada Gambar 3.12 sedangkan *pseudocode* ditunjukkan oleh Gambar 3.13, Gambar 3.14, Gambar 3.15, Gambar 3.16, dan Gambar 3.17. Perancangan yang sudah dilakukan pada bab ini dilanjutkan dengan implementasi menjadi kode sumber untuk membangun program. Implementasi tersebut dijelaskan lebih lanjut pada bab 4.

Input	Dataset NWFS (data), parameter populasi kromosom (populasi), jumlah banyaknya pekerjaan dari dataset (job), jumlah banyaknya mesin dari dataset (mesin)
Output	Urutan pekerjaan yang terbaik (bestKromosom), <i>Makespan</i> yang terbaik (bestWaktu)
1	Inisiasi parameter Set populasi, set m, set cr, set parTS, set iterasiTS, set batasIterasiTS, set alphaTS
2	Inisiasi Data data = xlsread(dataset.csv') mesin = size (data, 2) job = size(data, 1)
3	Pembangkitan populasi kromosom for i = 1 : populasi temp = randperm(job) for j = 1 : job kromosom (i, j) = temp (1, j) endfor endfor

Gambar 3.13 Pseudocode Umum Metode GA-TS (Bagian 1)

4	Menghitung <i>makespan</i> kromosom timeKromosom(populasi)=0; for i=1:populasi [waktu]= fitness2(data,kromosom(i, :),mesin,job) timeKromosom(i)=waktu; endfor
5	Transpose <i>makespan</i> timeKromosom2 = transpose(timeKromosom) [waktuNWFS urutan] = sort (timeKromosom2)
6	Mengurutkan kromosom sesuai <i>makespan</i> secara ascending for i = 1 : populasi parent(i,:) = kromosom(urutan(i, 1),:) endfor [waktuNWFS urutan] = sort (waktuNWFS);
7	Pengambilan sampel kromosom berdasarkan <i>makespan</i> kecil for i = 1 : parTS kromosomTS(i,:) = parent(i,:) waktuTS(i) = waktuNWFS(i,1) endfor
8	Pengambilan sampel kromosom berdasarkan <i>makespan</i> sedang middleTS = round(populasi/2) - floor(parTS/2) for i = parTS+1 : parTS*2 kromosomTS(i,:) = parent(middleTS,:) waktuTS(i) = waktuNWFS(middleTS,1) middleTS = middleTS + 1
9	Pengambilan sampel kromosom berdasarkan <i>makespan</i> besar downTS = populasi+1 - parTS for i = parTS*2 + 1 : parTS*3 kromosomTS(i,:) = parent(downTS,:) waktuTS(i) = waktuNWFS(downTS,1) downTS = downTS + 1 endfor

Gambar 3.14 Pseudocode Umum Metode GA-TS (Bagian 2)

10	<p>Proses Tabu Search</p> <pre> while (iterasiTS > batasIterasiTS) jumlahMutasi = size(kromosomTS, 1) for i=1:jumlahMutasi posisi = randperm(job,2) p1 = min (posisi) p2 = max (posisi) tempKromosomTS(i,:) = kromosomTS(i,:) mutasiTemp = tempKromosomTS(i,p1); tempKromosomTS(i,p1) = tempKromosomTS(i,p2) tempKromosomTS(i,p2) = mutasiTemp; [waktuMutasi(i)] = fitness2(data,tempKromosomTS(i,:),mesin,job) if waktuMutasi(1, i) <= waktuTS(1,i) kromosomTS(i,:) = tempKromosomTS(i,:) else TS = exp((- (waktuMutasi(1,i) - waktuTS(1,i)))/iterasiTS) y = rand if TS >= y kromosomTS(i,:) = tempKromosomTS(i,:) endif endif endfor iterasiTS = iterasiTS * alphaTS ^ ii endwhile cc = populasi+1 waktuMutasi1 = transpose (waktuMutasi) </pre>
----	---

Gambar 3.15 Pseudocode Umum Metode GA-TS (Bagian 3)

11	Penambahan kromosom TS kedalam populasi kromosom awal for i = 1 : size(kromosomTS, 1) parent (cc, :) = kromosomTS(i,:) waktuNWFS (cc, 1) = waktuMutasi1(i, 1) cc = cc + 1 endfor
12	Menyeleksi kromosom terbaik berdasarkan banyaknya populasi for i = 1 : populasi parent(i,:) = parent(urutan(i,1),:) endfor
13	Menyeleksi kromosom untuk kawin silang for i = 1 : populasi r1(i) = rand endfor availCR = find (r1 < cr)
14	Proses Kawin Silang for i = 1 : iterCR if i == populasi [child1 child2] = crossover(parent(i,:), parent(1,:)) else [child1 child2] = crossover(parent(i,:), parent(i+1,:))
15	Menghitung makespan dari setiap child [waktuChild1] = fitness2(data,child1,mesin,job) [waktuChild2] = fitness2(data,child2,mesin,job)
16	Menyeleksi child terbaik berdasarkan makespan yang terkecil if waktuChild1 < waktuChild2 child(i,:) = child1 else child(i,:) = child2 endifor endfor

Gambar 3.16 Pseudocode Umum Metode GA-TS (Bagian 4)

17	Menyeleksi kromosom untuk melakukan mutasi for i=1 : size(child,1) r2(i) = rand endfor availM = find (r2 < m) iterM = size(availM,2)-1
18	Melakukan proses mutasi for i = 1 : iterM posisi = randperm(job,2) p1 = min (posisi) p2 = max (posisi) mutasiTemp = child (availM(i),p1) child (availM(i),p1) = child (availM(i),p2) child (availM(i),p2) = mutasiTemp endfor
19	Menggabungkan populasi awal dengan child merge = [parent;child]
20	Menghitung makespan kromosom gabungan for i=1:size(merge,1) [waktu]= fitness2(data,merge(i,:),mesin,job) waktuKromosomFinal(i)=waktu endfor
21	Mengurutkan kromosom berdasarkan makespan secara ascending waktuKromosomFinal2 = transpose(waktuKromosomFinal) [bestWaktu urutan] = sort (waktuKromosomFinal2)
22	Mengeluarkan hasil kromosom dan makespan terbaik disp('Best Sequence No-Wait Flow-Shop') disp(bestKromosom(1,:)) disp('Makespan No-Wait Flow-Shop') disp(bestWaktu(1,1))

Gambar 3.17 Pseudocode Umum Metode GA-TS (Bagian 5)

BAB IV IMPLEMENTASI

Bab ini berisi penjelasan mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun program. Kode sumber secara lengkap dapat dilihat pada Lampiran A sedangkan pada bab ini menjelaskan mengenai implementasi pada kode sumber mengenai cara untuk mendapatkan urutan pekerjaan terbaik dan mendapatkan nilai *makespan* terbaik. Implementasi menggunakan perangkat lunak MATLAB.

4.1 Lingkungan Implementasi

Implementasi penyelesaian pada permasalahan NWFS dengan menggunakan metode GA-TS menggunakan spesifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i3-3240 CPU @ 3.40 Ghz
	Memori	4 GB
Perangkat Lunak	Sistem Operasi	Microsoft Windows 8.1 Pro 64-bit
	Perangkat Pengembang	MATLAB 7.13.0.564 R2011b

4.2 Implementasi Metode GA-TS Untuk Mendapatkan Urutan Pekerjaan Terbaik

Pada subbab ini dijelaskan mengenai implementasi metode GA-TS untuk mendapatkan urutan pekerjaan terbaik.

Implementasi dilakukan berdasarkan perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi-implementasi yang dilakukan adalah impelementasi pembangkitan kromosom, pengambilan sampel kromosom untuk metode TS, proses metode TS, proses metode GA, dan pengurutan kromosom berdasarkan *makespan* terbaik.

4.2.1 Implementasi Pembangkitan Kromosom

Subbab ini menjelaskan mengenai implementasi pembangkitan kromosom berdasarkan banyaknya populasi. Pembangkitan kromosom atau urutan kerja dilakukan secara *random permutation*. Kromosom yang dibangkitkan berdasarkan pada data uji dan memiliki urutan-urutan kerja yang berbeda, seperti yang ditunjukkan oleh Tabel 2.3. Implementasi pembangkitan kromosom terdapat pada fungsi *main.m* dan dapat dilihat pada Kode Sumber 4.1.

1	<code>%%--Data</code>
2	<code>data = xlsread('J7M5.csv');</code>
3	<code>mesin = size (data, 2);</code>
4	<code>job = size(data, 1);</code>
5	<code>populasi = 70;</code>
6	<code>%%--Membangkitkan populasi</code>
7	<code>for i = 1 : populasi</code>
8	<code>temp = randperm(job);</code>
9	<code>for j = 1 : job</code>
10	<code>kromosom (i, j) = temp (1, j);</code>
11	<code>end</code>
12	<code>end</code>

Kode Sumber 4.1 Implementasi Pembangkitan Kromosom

4.2.2 Implementasi Pengambilan Sampel Kromosom Untuk Metode TS

Metode TS digunakan agar hasil pada metode GA tidak terjebak pada *local optima*. Kromosom-kromosom yang diproses pada metode TS tidak diambil semua, melainkan hanya beberapa kromosom yang diambil untuk menjadi sampel. Kromosom-kromosom diambil berdasarkan tiga kelompok yang sudah diklasifikasikan berdasarkan nilai *makespan*. Terdapat parameter sebagai nilai batas untuk pengambilan sampel. Semakin besar nilai tersebut, maka semakin banyak sampel yang dapat diambil. Parameter tersebut adalah *parTS*. Implementasi pembangkitan sampel kromosom untuk metode TS terdapat pada fungsi *main.m* dan dapat dilihat pada Kode Sumber 4.2 dan Kode Sumber 4.3.

1	<code>%%--Parameter</code>
2	<code>parTS = round(0.1*populasi);</code>
3	<code>populasi = 70;</code>
4	<code>kromosomTS = zeros(parTS*3, size(kromosom,2));</code>
5	<code>%%--Makespan Kecil</code>
6	<code>for i = 1 : parTS</code>
7	<code> kromosomTS(i,:) = parent(i,:)</code>
8	<code> waktuTS(i) = waktuNWFS(i,1)</code>
9	<code>end</code>
10	<code>%%--Makespan Sedang</code>
11	<code>middleTS = round(populasi/2) - floor(parTS/2);</code>
12	<code>for i = parTS+1 : parTS*2</code>
13	<code> kromosomTS(i,:) = parent(middleTS,:)</code>
14	<code> waktuTS(i) = waktuNWFS(middleTS,1)</code>

Kode Sumber 4.2 Implementasi Pengambilan Sampel Kromosom Untuk Metode TS (Bagian 1)

15	<code>middleTS = middleTS + 1;</code>
16	<code>end</code>
17	<code>%%--Makespan Besar</code>
18	<code>downTS = populasi+1 - parTS;</code>
19	<code>for i = parTS*2 + 1 : parTS*3</code>
20	<code> kromosomTS(i,:) = parent(downTS,:)</code>
21	<code> waktuTS(i) = waktuNWFS(downTS,1)</code>
22	<code> downTS = downTS + 1;</code>
23	<code>end</code>

**Kode Sumber 4.3 Implementasi Pengambilan Sampel
Kromosom Untuk Metode TS (Bagian 2)**

4.2.3 Implementasi Proses Metode TS

Implementasi metode TS dilakukan sebelum GA agar menghindari hasil yang didapat oleh GA terjebak di *local optima*. Metode TS menggunakan proses mutasi yang dijelaskan lebih lanjut pada subbab 4.2.4.2. Proses iterasi pada metode TS berhenti ketika nilai *iterasiTS* kurang dari nilai *batasIterasiTS*. Penurunan nilai iterasi tersebut sudah dijelaskan sebelumnya pada Persamaan 2.24. Setelah didapatkan hasil dari proses TS, maka kromosom dan *makespan* hasil dari mutasi pada TS ditambahkan kedalam populasi awal. Implementasi proses metode TS terdapat pada fungsi *main.m* dan dapat dilihat pada Kode Sumber 4.4, Kode Sumber 4.5, dan Kode Sumber 4.6.

1	<code>%%--Parameter</code>
2	<code>m = 0.001; %%--Mutation rate</code>
3	<code>iterasiTS = 25;</code>
4	<code>batasIterasiTS = 0.1;</code>
5	<code>alphaTS = 0.7;</code>
6	<code>%%--Proses TS</code>

Kode Sumber 4.4 Implementasi Proses Metode TS (Bagian 1)

7	<code>while (iterasiTS > batasIterasiTS)</code>
8	<code> %%--Mutasi TS</code>
9	<code> jumlahMutasi = size(kromosomTS, 1);</code>
10	<code> for i=1:jumlahMutasi</code>
11	<code> posisi = randperm(job,2);</code>
12	<code> p1 = min (posisi);</code>
13	<code> p2 = max (posisi);</code>
14	<code> tempKromosomTS(i,:) =</code> <code> kromosomTS(i,:);</code>
15	<code> mutasiTemp = tempKromosomTS(i,p1);</code>
16	<code> tempKromosomTS(i,p1) =</code> <code> tempKromosomTS(i,p2);</code>
17	<code> tempKromosomTS(i,p2) = mutasiTemp;</code>
18	<code> %%--Perhitungan Nilai Makespan</code>
19	<code> [waktuMutasi(i)] =</code> <code> fitness2(data,tempKromosomTS(i,:),mesin,job</code> <code>);</code>
20	<code> if waktuMutasi(1, i) <=</code> <code> waktuTS(1,i)</code>
21	<code> kromosomTS(i,:) =</code> <code> tempKromosomTS(i,:)</code>
22	<code> else</code>
23	<code> %%--Probabilitas Menjadi</code> <code> Kromosom Baru</code>
24	<code> TS = exp((- (waktuMutasi(1,i) -</code> <code> waktuTS(1,i))) / iterasiTS)</code>
25	<code> y = rand;</code>
26	<code> if TS >= y</code>
27	<code> kromosomTS(i,:) =</code> <code> tempKromosomTS(i,:)</code>
28	<code> end</code>
29	<code> end</code>

Kode Sumber 4.5 Implementasi Proses Metode TS (Bagian 2)

30	<code>end</code>
31	<code>iterasiTS = iterasiTS * alphaTS ^ ii</code>
32	<code>end</code>
33	<code>%%--Penggabungan Kromosom Hasil TS ke Populasi</code>
34	<code>cc = populasi+1;</code>
35	<code>waktuMutasi1 = transpose (waktuMutasi);</code>
36	<code>for i = 1 : size(kromosomTS, 1)</code>
37	<code>parent (cc, :) = kromosomTS(i, :);</code>
38	<code>waktuNWFS (cc, 1) = waktuMutasi1(i, 1);</code>
39	<code>cc = cc + 1;</code>
40	<code>end</code>

Kode Sumber 4.6 Implementasi Proses Metode TS (Bagian 3)

4.2.4 Implementasi Proses GA

Implementasi pada proses GA terbagi menjadi dua subbab bagian, yaitu implementasi proses kawin silang dan implementasi proses mutasi. Kedua proses tersebut merupakan proses utama pada GA. Proses kawin silang pada GA dijelaskan pada subbab 4.2.4.1 sedangkan proses mutasi pada GA dijelaskan pada subbab 4.2.4.2.

4.2.4.1 Implementasi Proses Kawin Silang

Implementasi proses kawin silang dilakukan berdasarkan perancangan proses kawin silang yang terdapat pada subbab 3.2.4.1. Implementasi proses kawin silang terdapat pada dua fungsi, yaitu *main.m* dan *crossover.m*. Fungsi pada *crossover.m* dipanggil dari fungsi *main.m*. Implementasi pada fungsi *main.m* dapat dilihat pada Kode Sumber 4.7 sedangkan implementasi pada fungsi *crossover.m* dapat dilihat pada Kode Sumber 4.8 dan Kode

Sumber 4.9. Proses irisan dapat dilihat pada Kode Sumber 4.10 dan Kode Sumber 4.11.

1	<code>%%--main.m</code>
2	<code>%%--Mencari kromosom untuk di Kawin Silang</code>
3	<code>for i = 1 : populasi</code>
4	<code> r1(i) = rand;</code>
5	<code>end</code>
6	<code>end</code>
7	<code>availCR = find (r1 < cr);</code>
8	<code>iterCR = size(availCR,2);</code>
9	<code>%%--Proses Kawin Silang</code>
10	<code>for i = 1 : iterCR</code>
11	<code> if i == populasi</code>
12	<code> [child1 child2] =</code> <code> crossover(parent(i,:), parent(1,:));</code>
13	<code> else</code>
14	<code> [child1 child2] =</code> <code> crossover(parent(i,:), parent(i+1,:));</code>
15	<code> end</code>
16	<code>%%--Hitung makespan setiap child</code>
17	<code> [waktuChild1] =</code> <code> fitness2(data,child1,mesin,job);</code>
18	<code> [waktuChild2] =</code> <code> fitness2(data,child2,mesin,job);</code>
19	<code>%%--Seleksi child yang terbaik</code>
20	<code> if waktuChild1 < waktuChild2</code>
21	<code> child(i,:) = child1;</code>
22	<code> else</code>
23	<code> child(i,:) = child2;</code>
24	<code> end</code>
25	<code>end</code>

Kode Sumber 4.7 Proses Kawin Silang Pada Fungsi *main.m*

1	<code>function [child1 child2] = crossover(parent1,parent2)</code>
2	<code>%%--Inisialisasi Variabel</code>
3	<code>counter1 = 1;</code>
4	<code>counter2 = 1;</code>
5	<code>%%--Panjang kromosom</code>
6	<code>length = size(parent1, 2);</code>
7	<code>%%--Operator Posisi</code>
8	<code>posisi = randperm (length-1, 2);</code>
9	<code>p1 = min (posisi);</code>
10	<code>if p1 == 1</code>
11	<code> p1 = p1 + 1;</code>
12	<code>end</code>
13	<code>p2 = max (posisi);</code>
14	<code>if p2 == p1</code>
15	<code> p2 = p2 + 1;</code>
16	<code>end</code>
17	<code>%%--Copy kromosom diantara posisi kawin</code>
18	<code>child1 = zeros (1, length);</code>
19	<code>child2 = zeros (1, length);</code>
20	<code>for j = p1 : p2</code>
21	<code> child1(j) = parent1(j);</code>
22	<code> child2(j) = parent2(j);</code>
23	<code>end</code>
24	<code>%%--Mencari kromosom yang beda</code>
25	<code>[tempc tempd] = slice(parent1, child1, parent2, child2, p1, p2)</code>
26	<code>%%--Memasukkan kromosom yang belakang</code>
27	<code>for j = p2 + 1 : length</code>
28	<code> child1(j) = tempc(counter1);</code>

**Kode Sumber 4.8 Implementasi Proses Kawin Silang Pada
Fungsi *crossover.m* (Bagian 1)**

29	child2(j) = tempd(counter2);
30	counter1 = counter1 + 1;
31	counter2 = counter2 + 1;
32	end
33	%%--Memasukkan kromosom yang depan
34	for j = 1 : p1 - 1
35	child1(j) = tempc(counter1);
36	child2(j) = tempd(counter2);
37	counter1 = counter1 + 1;
38	counter2 = counter2 + 1;
39	end
40	end

Kode Sumber 4.9 Implementasi Proses Kawin Silang Pada Fungsi *crossover.m* (Bagian 2)

1	function [tempc tempd] = slice(parent1, child1, parent2, child2, c1, c2)
2	%%--Memisahkan kromosom yang berbeda
3	[a b] = setxor (parent2, child1);
4	[d e] = setxor (parent1, child2);
5	%%--Sorting urutan
6	c = sort(b);
7	f = sort(e);
8	%%--Mengurutkan sesuai rules Crossover
9	for i = 1: banyak
10	if c(banyak) > c2
11	c = circshift (c, [0,1]);
12	end
13	if f(banyak) > c2

Kode Sumber 4.10 Implementasi Proses Irisan Pada Fungsi *irisan.m* (Bagian 1)

14	<code>f = circshift (f,[0,1]);</code>
15	<code>end</code>
16	<code>end</code>
17	<code>%%--Mengurutkan kromosom sesuai urutan</code>
18	<code>for i = 1 : size (c,2)</code>
19	<code>tempa = c (i);</code>
20	<code>tempb = f (i);</code>
21	<code>tempc (i) = parent2 (tempa);</code>
22	<code>tempd (i) = parent1 (tempb);</code>
23	<code>end</code>

Kode Sumber 4.11 Implementasi Proses Irisan Pada Fungsi *irisan.m* (Bagian 2)

4.2.4.2 Implementasi Proses Mutasi

Implementasi proses mutasi pada subbab ini berdasarkan perancangan proses mutasi yang dilakukan pada subbab 3.2.4.2. Proses mutasi digunakan dua kali pada proses TS dan proses GA. Mutasi pada TS tidak perlu diseleksi karena sudah terjadi penyeleksian pada saat pengambilan sampel kromosom sedangkan mutasi pada GA dilakukannya seleksi pada kromosom yang dapat dimutasi. Implementasi proses mutasi terdapat pada fungsi *main.m* dan dapat dilihat pada Kode Sumber 4.12 dan Kode Sumber 4.13.

1	<code>%%--Mencari Child untuk di Mutasi</code>
2	<code>for i=1 : size(child,1)</code>
3	<code> r2(i) = rand;</code>
4	<code>end</code>
5	<code>availM = find (r2 < m);</code>
6	<code>iterM = size(availM,2)-1;</code>
7	<code>%%--Proses Mutasi</code>

Kode Sumber 4.12 Implementasi Proses Mutasi (Bagian 1)

8	<code>for i = 1 : iterM</code>
9	<code>posisi = randperm(job,2);</code>
10	<code>p1 = min (posisi);</code>
11	<code>p2 = max (posisi);</code>
12	<code>mutasiTemp = child (availM(i),p1);</code>
13	<code>child (availM(i),p1) = child (availM(i),p2);</code>
14	<code>child (availM(i),p2) = mutasiTemp;</code>
15	<code>end</code>

Kode Sumber 4.13 Implementasi Proses Mutasi (Bagian 2)

4.2.5 Impelementasi Pengurutan Kromosom Berdasarkan *Makespan* Terbaik

Fungsi tujuan dari permasalahan NWFS adalah untuk mendapatkan nilai *makespan* yang terbaik. Untuk memudahkan proses pada metode TS dan metode GA dan memudahkan untuk mengetahui *makespan* dan juga kromosom terbaik yang ada, maka kromosom diurutkan berdasarkan nilai *makespan* yang terbaik. Implementasi pengurutan kromosom berdasarkan *makespan* terbaik terdapat pada fungsi *main.m* dan dapat dilihat pada Kode Sumber 4.14.

1	<code>%%--Mengurutkan Makespan Ascending</code>
2	<code>[waktuNWFS urutan] = sort (timeKromosom);</code>
3	<code>%%--Mengurutkan Kromosom Sesuai Makespan</code>
4	<code>for i = 1 : populasi</code>
5	<code>parent(i,:) = parent(urutan(i,1),:);</code>
6	<code>end</code>

Kode Sumber 4.14 Implementasi Pengurutan Kromosom Berdasarkan *Makespan* Terbaik

4.3 Implementasi Metode GA-TS Untuk Mendapatkan Nilai *Makespan*

Pada subbab ini dijelaskan mengenai implementasi metode GA-TS untuk mendapatkan nilai *makespan* pada permasalahan NWFS. Implementasi dilakukan berdasarkan perancangan pada subbab 3.3. Penjelasan mengenai implementasi metode GA-TS untuk mendapatkan nilai *makespan* dijelaskan pada subbab 4.3.1.

4.3.1 Implementasi Perhitungan Nilai *Makespan*

Subbab ini menjelaskan mengenai perhitungan untuk mendapatkan nilai *makespan*. Implementasi ini bertujuan untuk mendapatkan waktu seoptimal mungkin pada permasalahan penjadwalan NWFS. Selain menghitung nilai *makespan*, implementasi ini juga menghitung waktu tunggu yang terdapat pada setiap pekerjaan. Waktu tunggu tersebut berguna untuk menghitung berapa lama pekerjaan diproses setelah pekerjaan sebelumnya selesai diproses. Implementasi perhitungan nilai *makespan* terdapat pada fungsi *fitness.m*. Fungsi pada *fitness.m* dipanggil dari fungsi *main.m*. Implementasi pada *fitness.m* dapat dilihat pada Kode Sumber 4.15, Kode Sumber 4.16, dan Kode Sumber 4.17.

1	<code>%%--Waktu Tunggu</code>
2	<code>for i = 1 : job</code>
3	<code> for k = 1 : job</code>
4	<code> if i == k</code>
5	<code> delay(i,k) = 0;</code>
6	<code> else</code>
7	<code> temp = data(i,2) - data(k,1);</code>

Kode Sumber 4.15 Implementasi Perhitungan Nilai *Makespan* (Bagian 1)

8	temp2 = sum(data(i,2:mesin),2) - sum(data(k,1:mesin-1),2);
9	compare = [temp2 temp 0];
10	delay(i,k) = max(compare);
11	end
12	end
13	end
14	%%--Pemindahan Data Sesuai Urutan Kromosom
15	for j=1 : job
16	for i=1 : mesin
17	dataKromosom (j,i) = data(kromosom(1, j), i);
18	end
19	end
20	%%--Nilai Proses Pada Pekerjaan Pertama
21	for j = 1 : job
22	if j==1
23	for i = 1 : mesin
24	if i == 1
25	waktu(j, i) = dataKromosom (j,i);
26	else
27	waktu (j, i) = waktu(j, i-1) + dataKromosom (j,i);
28	end
29	end
30	end
31	end
32	%%--Waktu Tunggu Pada Mesin Pertama
33	for j=2:job

Kode Sumber 4.16 Implementasi Perhitungan Nilai *Makespan* (Bagian 2)

34	waktu (j, 1) = waktu(j-1, 1) + delay(kromosom(1,j-1),kromosom(1,j)) + dataKromosom(j,1);
35	end
36	%%--Nilai Proses Sesuai Directed Graph
14	for j=2:job
15	for i = 2 : mesin
16	if waktu(j, i-1) > waktu (j-1, i)
17	waktu(j, i) = waktu(j, i-1) + dataKromosom (j,i);
18	else
19	waktu(j, i) = waktu(j-1, i) + dataKromosom (j,i);
20	end
21	end
22	end
20	%%--Makespan Terbaik
20	waktu = waktu(j,i);

Kode Sumber 4.17 Implementasi Perhitungan Nilai *Makespan* (Bagian 3)

BAB V

HASIL UJI COBA DAN EVALUASI

Bab ini berisi penjelasan mengenai uji coba dan evaluasi pada permasalahan penjadwalan NWFS dengan menggunakan metode GA-TS. Hasil uji coba didapatkan dari implementasi pada bab 4. Bab ini berisikan pembahasan mengenai lingkungan pengujian, data pengujian, dan uji kinerja.

5.1 Lingkungan Pengujian

Lingkungan pengujian pada uji coba permasalahan NWFS dengan menggunakan metode GA-TS menggunakan spesifikasi keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Pengujian

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel(R) Core(TM) i3-3240 CPU @ 3.40 Ghz
	Memori	4 GB
Perangkat Lunak	Sistem Operasi	Microsoft Windows 8.1 Pro 64-bit
	Perangkat Pengembang	MATLAB 7.13.0.564 R2011b

5.2 Data Pengujian

Subbab ini menjelaskan mengenai data yang digunakan pada uji coba. Data uji coba yang digunakan berasal dari Carlier *car1-car8* dan Reeves *rec01-rec41*. Data uji coba direpresentasikan dalam *file* dengan format *Comma Separated Values* (CSV). Data tersebut merupakan data NWFS yang berisikan waktu proses pada setiap pekerjaan dan setiap mesin. Jumlah pekerjaan dan jumlah

mesin adalah dua parameter yang sudah ditetapkan pada permasalahan NWFS ini. Data uji coba terbagi dalam dua kategori, yaitu data besar dan data kecil. Data kecil meliputi 7 hingga 20 pekerjaan dan 4 hingga 10 mesin, sedangkan data besar meliputi 20 hingga 75 pekerjaan dan 10 hingga 20 mesin.

5.3 Uji Kinerja

Subbab ini menguji kinerja program yang sudah dirancang dan diimplementasikan pada penyelesaian permasalahan NWFS dengan menggunakan metode GA-TS. Tujuan utama dari uji coba adalah untuk membandingkan *makespan* dari implementasi algoritma-algoritma NWFS yang sebelumnya sudah pernah diuji coba. Uji coba membandingkan metode GA-TS dengan algoritma *Genetic Algorithm* (GA) dari Chaudry & Khan dan hasil uji coba dari penyedia data uji coba, yaitu Carlier dan Reeves. Hasil uji coba yang didapat oleh Carlier dan Reeves dinamakan hasil penemu. Hasil uji coba menghasilkan nilai yang bervariasi karena sifat dari metode GA-TS yang menggunakan nilai acak. Oleh karena itu, hasil yang ditunjukkan selain nilai *makespan* terbaik terdapat juga persentase perbedaan antara hasil uji coba GA-TS dengan hasil terbaik. Nilai *makespan* terbaik diberikan label 'min' dan persentase perbedaan diberikan label '%diff'. Perhitungan persentase perbedaan nilai *makespan* tersebut sudah dijelaskan oleh Persamaan 2.13. Uji coba dilakukan sebanyak 50 kali pada setiap data uji coba. Nilai *makespan* yang dihasilkan berbentuk satuan t. Satuan t digunakan karena dapat diambil asumsi bahwa hasil bisa digunakan sebagai satuan detik, menit, atau jam. Uji coba ini memiliki pengaturan-pengaturan pada variabel tertentu agar mendapatkan hasil yang maksimal. Pengaturan-pengaturan tersebut meliputi jumlah uji coba yang dilakukan, jumlah kromosom pada populasi, parameter pada kawin silang, parameter pada mutasi, parameter iterasi pada TS, parameter batas iterasi pada TS, dan parameter *alpha* TS. Pengaturan tersebut ditunjukkan

pada Tabel 5.2. Hasil Uji coba lengkap dapat dilihat pada Lampiran B.

Tabel 5.2 Pengaturan Variabel Pada Data Uji Coba

Deskripsi Variabel	Nilai
Jumlah uji coba yang dilakukan	50
Jumlah kromosom populasi awal	70
Parameter Kawin Silang	0,85
Parameter Mutasi	0,001
Parameter Iterasi <i>Tabu Search</i>	25
Parameter Batas Iterasi <i>Tabu Search</i>	0.1
Parameter <i>alpha Tabu Search</i>	0.7

Data uji coba dibagi menjadi dua kategori, yaitu data kecil dan data besar. Data kecil ditunjukkan oleh Tabel 5.3 dan Tabel 5.4 sedangkan data besar ditunjukkan oleh Tabel 5.5.

Tabel 5.3 Data Uji Coba Kategori Kecil (Bagian 1)

Nama Data Uji Coba	Jumlah Pekerjaan dan Mesin
<i>car1.csv</i>	J11M5
<i>car2.csv</i>	J13M4
<i>car3.csv</i>	J12M5
<i>car4.csv</i>	J14M4
<i>car5.csv</i>	J10M6
<i>car6.csv</i>	J8M9
<i>car7.csv</i>	J7M7
<i>car8.csv</i>	J8M8
<i>rec01.csv</i>	J20M5
<i>rec03.csv</i>	J20M5
<i>rec05.csv</i>	J20M5
<i>rec07.csv</i>	J20M10

Tabel 5.4 Data Uji Coba Kategori Kecil (Bagian 2)

Nama Data Uji Coba	Jumlah Pekerjaan dan Mesin
<i>rec09.csv</i>	J20M10
<i>rec11.csv</i>	J20M10

Tabel 5.5 Data Uji Coba Kategori Besar

Nama Data Uji Coba	Jumlah Pekerjaan dan Mesin
<i>rec13.csv</i>	J20M15
<i>rec15.csv</i>	J20M15
<i>rec17.csv</i>	J20M15
<i>rec19.csv</i>	J30M10
<i>rec21.csv</i>	J30M10
<i>rec23.csv</i>	J30M10
<i>rec25.csv</i>	J30M15
<i>rec27.csv</i>	J30M15
<i>rec29.csv</i>	J30M15
<i>rec31.csv</i>	J50M10
<i>rec33.csv</i>	J50M10
<i>rec35.csv</i>	J50M10
<i>rec37.csv</i>	J75M20
<i>rec39.csv</i>	J75M20
<i>rec41.csv</i>	J75M20

Pada Tabel 5.3, Tabel 5.4, dan Tabel 5.5 terdapat kolom nama data uji coba dan jumlah pekerjaan dan mesin. Pada kolom nama data uji coba, data uji coba berupa berkas berbentuk CSV (*Comma Separated Value*) dan pada kolom jumlah pekerjaan dan Mesin, J menunjukkan jumlah pekerjaan dan M menunjukkan jumlah mesin. Sebagai contoh apabila tertulis J11M5, maka dibaca sebagai 11 jumlah pekerjaan dan 5 jumlah mesin.

5.3.1 Uji Kinerja Data Uji Coba Kategori Kecil

Pada subbab ini ditunjukkan hasil uji kinerja pada data uji coba kategori kecil yang sudah ditunjukkan pada Tabel 5.3 dan Tabel 5.4. Hasil pengujian dari metode GA-TS yang dibandingkan dengan metode-metode lainnya pada uji coba kategori kecil dapat dilihat pada Tabel 5.6

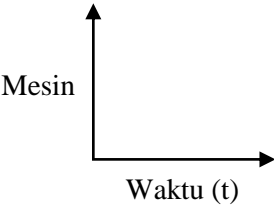
Tabel 5.6 Hasil Uji Coba Kategori Kecil

Data Uji Coba	n x m	Hasil Pene mu	GA (Chaudry & Khan)		GA-TS	
		Min	Min	%diff	Min	%diff
<i>car1.csv</i>	J11M5	8142	8142	0	7998	-1,77
<i>car2.csv</i>	J13M4	8242	8242	0	8758	6,26
<i>car3.csv</i>	J12M5	8866	8866	0	8760	-1,20
<i>car4.csv</i>	J14M4	9195	9195	0	9979	8,53
<i>car5.csv</i>	J10M6	9159	9159	0	8412	-8,16
<i>car6.csv</i>	J8M9	9690	9690	0	9101	-6,08
<i>car7.csv</i>	J7M7	7705	7705	0	7187	-6,72
<i>car8.csv</i>	J8M8	9372	9372	0	8879	-5,26
<i>rec01.csv</i>	J20M5	1590	1532	-3,65	1629	2,39
<i>rec03.csv</i>	J20M5	1457	1364	-6,38	1446	-0,75
<i>rec05.csv</i>	J20M5	1637	1520	-7,15	1517	-7,33
<i>rec07.csv</i>	J20M10	2119	2088	-1,46	2008	-5,24
<i>rec09.csv</i>	J20M10	2141	2058	-3,88	1922	-10,23
<i>rec11.csv</i>	J20M10	1946	1906	-2,06	1842	-5,34

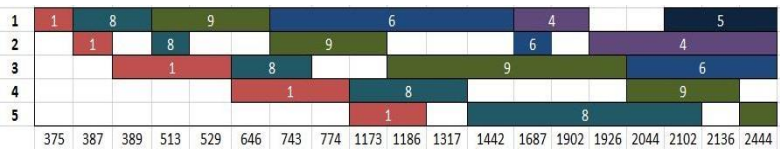
Pada Tabel 5.6 diketahui bahwa nilai *makespan* dari metode GA-TS mengungguli 80% dibandingkan hasil uji coba dari penyedia data sedangkan metode GA-TS mampu mengungguli 71% dibandingkan metode GA. Pada data berlabel *car2.csv*, *car4.csv*, dan *rec01.csv* metode GA-TS tidak dapat mengungguli hasil dari penyedia data uji coba sedangkan pada data berlabel *car2.csv*, *car4.csv*, *rec01.csv*, dan *rec03.csv* metode GA-TS tidak

dapat mengungguli metode GA. Kesimpulan yang dapat diambil adalah metode GA-TS dapat menghasilkan *makespan* yang terbaik, tetapi tidak dapat menghasilkan *makespan* yang optimal pada permasalahan yang memiliki 13, 14, dan 20 pekerjaan dengan 4 dan 5 mesin. Hasil uji coba selalu menghasilkan nilai *makespan* yang berbeda-beda karena nilai acak yang dibangkitkan.

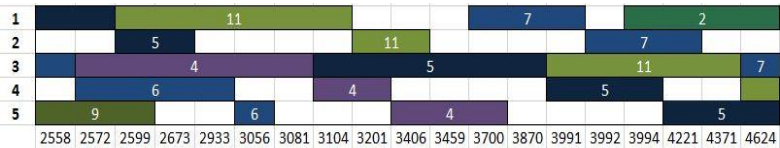
Hasil uji coba pada Tabel 5.6 dapat dibuktikan keselarasan dengan menggunakan *Gantt Chart*. Pembuktian dilakukan oleh beberapa data uji coba, yaitu *car1.csv*, *car8.csv*, *rec01.csv*, dan *rec11.csv*. Keterangan atribut pada *Gantt Chart* ditunjukkan oleh Gambar 5.1. Pembuktian keselarasan data uji *car1.csv* ditunjukkan oleh Gambar 5.2, Gambar 5.3, dan Gambar 5.4 sedangkan *dataset* dan pembuktian keselarasan data uji *car8.csv*, *rec01.csv*, dan *rec11.csv* ditunjukkan pada Lampiran D.



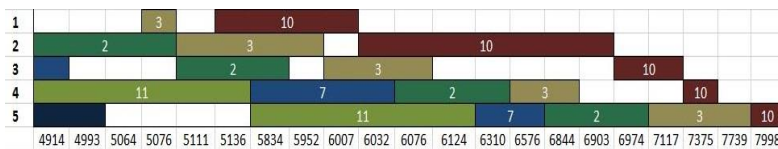
Gambar 5.1 Keterangan Atribut Pada *Gantt Chart*



Gambar 5.2 *Gantt Chart* car1.csv (Bagian 1)



Gambar 5.3 *Gantt Chart* car1.csv (Bagian 2)



Gambar 5.4 Gantt Chart car1.csv (Bagian 3)

Pada ketiga *Gantt Chart* tersebut dapat dilihat bahwa nilai *makespan* pada *Gantt Chart* dengan nilai *makespan* pada Tabel 5.6 sudah sesuai.

5.3.2 Uji Kinerja Data Uji Coba Kategori Besar

Pada subbab ini ditunjukkan hasil uji kinerja pada data uji coba kategori besar. Data uji coba kategori besar dapat dilihat pada Tabel 5.5. Data uji coba besar memiliki jumlah pekerjaan dan mesin yang beragam. Hasil pengujian dari metode GA-TS dibandingkan dengan metode-metode lainnya pada uji coba kategori besar dapat dilihat pada Tabel 5.7 dan Tabel 5.8.

Tabel 5.7 Hasil Uji Coba Kategori Besar (Bagian 1)

Data Uji Coba	n x m	Hasil Pene mu	GA (Chaudry & Khan)		GA-TS	
		Min	Min	%diff	Min	%diff
rec13.csv	J20M15	2709	2565	-5,32	2409	-11,07
rec15.csv	J20M15	2691	2565	-4,68	2453	-8,84
rec17.csv	J20M15	2740	2599	-5,15	2343	-14,49
rec19.csv	J30M10	3157	2952	-6,49	3062	-3,01
rec21.csv	J30M10	3015	2891	-4,11	2787	-7,56

Tabel 5.8 Hasil Uji Coba Kategori Besar (Bagian 2)

Data Uji Coba	n x m	Hasil Penguji	GA		GA-TS	
		Min	Min	%diff	Min	%diff
<i>rec23.csv</i>	J30M10	3030	2783	-8,15	2955	-2,48
<i>rec25.csv</i>	J30M15	3835	3662	-4,51	3483	-9,18
<i>rec27.csv</i>	J30M15	3655	3565	-2,46	3374	-7,69
<i>rec29.csv</i>	J30M15	3583	3383	-5,58	3345	-6,64
<i>rec31.csv</i>	J50M10	4631	4528	-2,22	4860	4,94
<i>rec33.csv</i>	J50M10	4770	4711	-1,24	4773	0,06
<i>rec35.csv</i>	J50M10	4718	4634	-1,78	4995	5,87
<i>rec37.csv</i>	J75M20	8979	9515	8,16	8045	-10,39
<i>rec39.csv</i>	J75M20	9158	9885	7,94	8225	-10,19
<i>rec41.csv</i>	J75M20	9344	9888	5,82	8933	-4,40

Pada Tabel 5.7 dan Tabel 5.8 diketahui bahwa nilai *makespan* dari metode GA-TS dapat mengungguli 80% dibandingkan hasil uji coba dari penyedia data dan metode GA-TS mampu mengungguli 67% dibandingkan metode GA. Metode GA-TS tidak dapat mengungguli nilai *makespan* dari hasil uji coba penyedia data pada data berlabel *rec31.csv*, *rec33.csv* dan *rec35.csv*. Metode GA-TS tidak dapat mengungguli metode GA-SA karena memiliki nilai *makespan* yang lebih besar pada data berlabel *rec19.csv*, *rec23.csv*, *rec31.csv*, *rec33.csv* dan *rec35.csv*.

Hasil uji coba selalu menghasilkan nilai *makespan* yang berbeda-beda karena nilai acak yang dibangkitkan. Kesimpulan yang dapat diambil adalah metode GA-TS dapat menghasilkan *makespan* yang terbaik walaupun data yang diberikan besar, tetapi kurang dapat memberikan nilai maksimal pada data yang memiliki 30 dan 50 pekerjaan.

5.4 Uji Perbandingan Parameter Metode GA-TS

Subbab ini melakukan uji coba berdasarkan beberapa parameter yang dianggap memiliki peranan penting terhadap penyelesaian permasalahan NWFS dengan menggunakan metode GA-TS. Tujuan uji perbandingan adalah untuk mengetahui parameter-parameter yang optimal karena metode metaheuristik GA-TS bergantung pada parameter yang digunakan. Uji perbandingan parameter dilakukan dengan cara membandingkan tiga skenario dengan nilai yang berbeda pada salah satu parameter yang diuji coba. Nilai parameter yang optimal digunakan pada uji coba perbandingan parameter selanjutnya. Parameter yang dibandingkan adalah parameter jumlah populasi, *tabu search*, batas iterasi *tabu search*, kawin silang, dan mutasi. Data uji coba yang digunakan ditunjukkan oleh Tabel 5.9. Hasil lengkap uji coba perbandingan dapat dilihat pada lampiran C.

Tabel 5.9 Data Uji Coba Perbandingan Parameter

Nama Data Uji Coba	Jumlah Pekerja dan Mesin
car1.csv	J11M5
car5.csv	J10M6
rec01.csv	J20M5
rec13.csv	J20M15
rec25.csv	J30M15
rec37.csv	J75M20

5.4.1 Uji Perbandingan Parameter Jumlah Populasi

Jumlah populasi merupakan salah satu parameter penting untuk mendapatkan nilai *makespan* yang terbaik. Jika parameter populasi berjumlah sedikit, maka kemungkinan untuk mendapatkan *makespan* terbaik juga menjadi lebih kecil. Sedangkan apabila populasi berjumlah banyak, maka kemungkinan untuk mendapatkan *makespan* terbaik juga menjadi lebih besar. Permasalahan yang terjadi adalah apabila populasi berjumlah banyak, maka membutuhkan waktu proses yang lama. Dibutuhkan jumlah populasi yang tepat agar mendapatkan *makespan* terbaik dan waktu proses yang tidak terlalu lama.

Uji perbandingan parameter jumlah populasi dibagi menjadi tiga skenario, yaitu populasi berjumlah 25, 70, dan 100 seperti yang ditunjukkan pada Tabel 5.10. Uji coba perbandingan diantara ketiga skenario tersebut dilakukan sebanyak 50 kali iterasi. Hasil uji coba perbandingan tersebut ditunjukkan pada Tabel 5.11 dan Tabel 5.12.

Tabel 5.10 Skenario Parameter Jumlah Populasi Yang Diuji Coba

	Skenario 1	Skenario 2	Skenario 3
Jumlah Populasi	25	70	100

Tabel 5.11 Perbandingan *Makespan* Pada Uji Perbandingan Parameter Jumlah Populasi (Bagian 1)

Data Uji Coba	Skenario 1 (25)	Skenario 2 (70)	Skenario 3 (100)
<i>car1.csv</i>	8235	7998	8050
<i>car5.csv</i>	8619	8412	8476
<i>rec01.csv</i>	1677	1629	1655
<i>rec13.csv</i>	2847	2409	2482
<i>rec25.csv</i>	3699	3483	3619

Tabel 5.12 Perbandingan *Makespan* Pada Uji Perbandingan Parameter Jumlah Populasi (Bagian 2)

Data Uji Coba	Skenario 1 (25)	Skenario 2 (70)	Skenario 3 (100)
<i>rec37.csv</i>	8296	8045	8296

Tabel 5.11 dan Tabel 5.12 menunjukkan bahwa skenario dengan jumlah populasi 70 menghasilkan nilai *makespan* yang terbaik. Nilai *makespan* yang didapatkan pada skenario pertama dengan jumlah populasi 10 belum menemukan nilai yang terbaik. Hal tersebut terjadi karena populasi yang ada terlalu sedikit untuk menemukan nilai *makespan* yang terbaik. Pada skenario ketiga dengan jumlah populasi 100, memakan waktu yang lebih lama dibandingkan dengan skenario kedua. Skenario kedua merupakan skenario terbaik dengan jumlah populasi 70, maka untuk uji coba selanjutnya parameter jumlah populasi yang digunakan adalah 70 populasi.

5.4.2 Uji Perbandingan Parameter Nilai Awal Iterasi *Tabu Search*

Nilai awal iterasi TS menentukan banyaknya iterasi yang dilakukan pada proses TS. Parameter ini berfungsi untuk menentukan nilai awal pada iterasi yang terjadi pada metode TS. Parameter ini terus menurun nilainya ketika iterasi terjadi seperti yang ditunjukkan pada Persamaan 2.24. Nilai parameter terus menurun hingga mencapai nilai batas iterasi TS seperti yang ditunjukkan pada Persamaan 2.23.

Uji perbandingan parameter nilai awal iterasi TS dibagi menjadi tiga skenario, yaitu dengan nilai 10, 25, dan 50 seperti yang ditunjukkan pada Tabel 5.13. Uji coba perbandingan diantara ketiga skenario tersebut dilakukan sebanyak 50 kali iterasi. Hasil uji coba perbandingan tersebut ditunjukkan pada Tabel 5.14.

Tabel 5.13 Skenario Parameter Nilai Awal Iterasi *Tabu Search* Yang Diuji Coba

	Skenario 1	Skenario 2	Skenario 3
Nilai Awal Iterasi <i>Tabu Search</i>	10	25	50

Tabel 5.14 Perbandingan *Makespan* Pada Uji Perbandingan Parameter Nilai Awal Iterasi *Tabu Search*

Data Uji Coba	Skenario 1 (10)	Skenario 2 (25)	Skenario 3 (50)
<i>car1.csv</i>	8067	7998	8126
<i>car5.csv</i>	8368	8412	8310
<i>rec01.csv</i>	1646	1629	1655
<i>rec13.csv</i>	2454	2409	2437
<i>rec25.csv</i>	3527	3483	3563
<i>rec37.csv</i>	8344	8045	8381

Tabel 5.14 menunjukkan nilai *makespan* pada uji perbandingan dengan menggunakan tiga skenario seperti yang ditunjukkan oleh Tabel 5.13. Skenario kedua mengungguli lima dari enam data uji coba yang disediakan. Oleh karena itu, parameter nilai awal iterasi TS dengan nilai 25 dapat dikatakan lebih baik dibandingkan dengan nilai lainnya, yaitu 10 dan 50. Rata-rata persentase perbedaan nilai *makespan* antara skenario kedua dengan skenario pertama adalah -1,35%. Persentase tersebut menunjukkan bahwa skenario kedua unggul dibandingkan dengan skenario pertama. Sedangkan rata-rata persentase perbedaan nilai *makespan* antara skenario kedua dan ketiga adalah -1,56%. Persentase tersebut menunjukkan bahwa skenario kedua unggul dibandingkan dengan skenario ketiga.

Kesimpulan yang didapatkan pada uji perbandingan parameter nilai awal iterasi TS adalah skenario kedua merupakan skenario terbaik dengan parameter nilai awal iterasi TS 25, maka parameter nilai awal iterasi TS yang digunakan adalah 25.

5.4.3 Uji Perbandingan Parameter Batas Iterasi *Tabu Search*

Pada subbab ini menguji coba tiga parameter batas iterasi TS. Parameter tersebut menentukan waktu selesai proses metode TS. Proses metode TS berhenti ketika nilai iterasi kurang dari batas iterasi seperti yang ditunjukkan oleh Persamaan 2.23.

Uji perbandingan parameter batas iterasi TS dibagi menjadi tiga skenario, yaitu dengan nilai 0,1, 0,5, dan 1,0 seperti yang ditunjukkan pada Tabel 5.15. Uji coba perbandingan diantara ketiga skenario tersebut dilakukan sebanyak 50 kali iterasi. Hasil uji coba perbandingan tersebut ditunjukkan pada Tabel 5.16.

Tabel 5.15 Skenario Parameter Batas Iterasi *Tabu Search* Yang Diuji Coba

	Skenario 1	Skenario 2	Skenario 3
Batas Iterasi <i>Tabu Search</i>	0,1	0,5	1,0

Tabel 5.16 Perbandingan *Makespan* Pada Uji Perbandingan Parameter Batas Iterasi *Tabu Search*

Data Uji Coba	Skenario 1 (0,1)	Skenario 2 (0,5)	Skenario 3 (1,0)
<i>car1.csv</i>	7998	8114	8032
<i>car5.csv</i>	8412	8489	8412
<i>rec01.csv</i>	1629	1679	1646
<i>rec13.csv</i>	2409	2500	2482
<i>rec25.csv</i>	3483	3555	3638
<i>rec37.csv</i>	8045	8400	8284

Ditunjukkan pada Tabel 5.16 bahwa dari enam percobaan pada uji perbandingan yang sudah dilakukan, skenario pertama mengungguli dua skenario lainnya. Parameter batas iterasi TS 0,1 lebih baik dibandingkan dengan parameter lainnya, yaitu 0,5 dan 1,0. Rata-rata persentase perbedaan nilai *makespan* antara skenario

pertama dengan skenario kedua adalah -2,54%. Persentase tersebut menunjukkan bahwa skenario pertama unggul dibandingkan dengan skenario kedua. Sedangkan rata-rata persentase perbedaan nilai *makespan* antara skenario pertama dan ketiga adalah -1,93%. Persentase tersebut menunjukkan bahwa skenario pertama unggul dibandingkan dengan skenario ketiga. Kesimpulan yang didapatkan pada uji perbandingan parameter batas iterasi TS ini adalah skenario pertama merupakan skenario terbaik dengan parameter batas iterasi TS 0,1, maka untuk uji coba selanjutnya batas iterasi TS yang digunakan adalah 0,1.

5.4.4 Uji Perbandingan Parameter Kawin Silang

Kawin silang merupakan salah satu hal penting dalam metode algoritma gentika, oleh karena itu parameter kawin silang juga menjadi salah satu hal yang penting dalam metode algoritma genetika. Parameter kawin silang menentukan jumlah kromosom yang mengalami kawin silang. Semakin besar nilai parameter kawin silang, maka semakin banyak kromosom yang mengalami kawin silang. Semakin kecil nilai parameter kawin silang, maka semakin sedikit kromosom yang mengalami kawin silang. Seperti yang dikemukakan oleh Wen-Yang Ling, Wen-Yuan Lee, dan Tzung-Pei Hong [8], bahwa nilai parameter kawin silang lebih baik berkisar diantara 0,5 hingga 1,0.

Tabel 5.17 Skenario Parameter Kawin Silang Yang Diuji Coba

	Skenario 1	Skenario 2	Skenario 3
Kawin Silang	0,65	0,85	0,90

Uji perbandingan parameter kawin silang dibagi menjadi tiga skenario, yaitu parameter kawin silang dengan nilai 0,65, 0,85, dan 0,90 seperti yang ditunjukkan pada Tabel 5.17. Uji coba perbandingan diantara ketiga skenario tersebut dilakukan

sebanyak 50 kali iterasi. Hasil uji coba perbandingan tersebut ditunjukkan pada Tabel 5.18.

Tabel 5.18 Perbandingan *Makespan* Pada Uji Perbandingan Parameter Kawin Silang

Data Uji Coba	Skenario 1 (0,65)	Skenario 2 (0,85)	Skenario 3 (0,90)
<i>car1.csv</i>	8066	7998	8107
<i>car5.csv</i>	8628	8412	8343
<i>rec01.csv</i>	1671	1629	1631
<i>rec13.csv</i>	2445	2409	2430
<i>rec25.csv</i>	3630	3483	3633
<i>rec37.csv</i>	8542	8045	8197

Ditunjukkan pada Tabel 5.18 bahwa dari enam percobaan pada uji perbandingan yang sudah dilakukan, skenario kedua mengungguli dua skenario lainnya. Parameter kawin silang 0,85 lebih baik dibandingkan dengan parameter kawin silang 0,65 dan 0,90. Rata-rata persentase perbedaan nilai *makespan* antara skenario kedua dengan skenario pertama adalah -2,88%. Persentase tersebut menunjukkan bahwa skenario kedua unggul dibandingkan dengan skenario pertama. Sedangkan rata-rata persentase perbedaan nilai *makespan* antara skenario kedua dan ketiga adalah -1,26%. Persentase tersebut menunjukkan bahwa skenario kedua unggul dibandingkan dengan skenario ketiga.

Kesimpulan yang didapatkan pada uji perbandingan parameter kawin silang ini adalah skenario kedua merupakan skenario terbaik dengan parameter kawin silang 0,85, maka untuk uji coba selanjutnya parameter kawin silang yang digunakan adalah 0,85.

5.4.5 Uji Perbandingan Parameter Mutasi

Parameter mutasi merupakan salah satu hal yang penting di dalam metode GA-TS karena proses mutasi terdapat pada kedua

metode, yaitu mutasi pada Algoritma Genetika dan mutasi pada *Tabu Search*. Parameter mutasi menentukan banyaknya kromosom yang dimutasi. Semakin kecil parameter mutasi, maka semakin kecil kemungkinan kromosom dimutasi sedangkan semakin besar parameter mutasi, maka semakin besar kemungkinan kromosom dimutasi. Parameter mutasi dianjurkan untuk memiliki nilai yang kecil karena apabila banyak kromosom yang terkena mutasi menyebabkan didapatkannya bentuk kromosom yang sama. Nilai parameter mutasi lebih baik berkisar diantara 0,001 hingga 0,05 seperti yang dikemukakan oleh Wen-Yang Ling, Wen-Yuan Lee, dan Tzung-Pei Hong [8].

Tabel 5.19 Skenario Parameter Mutasi Yang Diuji Coba

	Skenario 1	Skenario 2	Skenario 3
Mutasi	0,001	0,005	0,05

Uji perbandingan parameter mutasi dibagi menjadi tiga skenario, yaitu parameter mutasi dengan nilai 0,001, 0,005, dan 0,05 seperti yang ditunjukkan pada Tabel 5.19. Uji coba perbandingan diantara ketiga skenario tersebut dilakukan sebanyak 50 kali iterasi. Hasil uji coba perbandingan tersebut ditunjukkan pada Tabel 5.20.

Tabel 5.20 Perbandingan *Makespan* Pada Uji Perbandingan Parameter Mutasi

Data Uji Coba	Skenario 1 (0,001)	Skenario 2 (0,005)	Skenario 3 (0,05)
<i>car1.csv</i>	7998	8154	8107
<i>car5.csv</i>	8412	8454	8466
<i>rec01.csv</i>	1629	1646	1636
<i>rec13.csv</i>	2409	2505	2409
<i>rec25.csv</i>	3483	3569	3644
<i>rec37.csv</i>	8045	8051	8225

Tabel 5.20 .menunjukkan nilai *makespan* pada uji perbandingan dengan menggunakan tiga skenario seperti yang ditunjukkan oleh Tabel 5.19. Skenario pertama mengungguli semua data uji coba yang disediakan. Oleh karena itu, parameter mutasi dengan nilai 0,001 dapat dikatakan lebih baik dibandingkan parameter mutasi dengan nilai 0,005 dan 0,05. Rata-rata persentase perbedaan nilai *makespan* antara skenario satu dengan skenario dua adalah -1,64%. Persentase tersebut menunjukkan bahwa skenario pertama unggul dibandingkan dengan skenario kedua. Sedangkan rata-rata persentase perbedaan nilai *makespan* antara skenario pertama dan ketiga adalah -1,51%. Persentase tersebut menunjukkan bahwa skenario pertama unggul dibandingkan dengan skenario ketiga.

Kesimpulan yang didapatkan pada uji perbandingan parameter mutasi adalah skenario pertama merupakan skenario terbaik dengan parameter mutasi 0,001, maka parameter mutasi yang digunakan adalah 0,001.

5.5 Uji Perbandingan Metode GA Dengan Metode GA-TS

Subbab ini melakukan uji perbandingan hasil *makespan* antara metode heuristik GA dengan metode metaheuristik GA-TS. Tujuan uji perbandingan adalah untuk mengetahui seberapa besar perbedaan hasil antara kedua metode tersebut. Diketahui sebelumnya bahwa penggunaan TS adalah untuk menghindari metode GA dari kemungkinan terjebak pada kondisi *local optima*. Data uji yang digunakan adalah data uji kecil dan data uji besar. Hasil uji perbandingan ditunjukkan oleh Tabel 5.21 dan Tabel 5.22.

Tabel 5.21 Hasil Uji Perbandingan Metode GA dengan Metode GA-TS (Bagian 1)

Data Uji Coba	n x m	GA	GA-TS	
		min	min	%diff
<i>car1.csv</i>	J11M5	8210	7998	-2,58%
<i>car2.csv</i>	J13M4	9256	8758	-5,38%

**Tabel 5.22 Hasil Uji Perbandingan Metode GA dengan
Metode GA-TS (Bagian 2)**

Data Uji Coba	n x m	GA	GA-TS	
		min	min	%diff
<i>car3.csv</i>	J12M5	9963	8760	-12,0%
<i>car4.csv</i>	J14M4	10080	9979	-1,00%
<i>car5.csv</i>	J10M6	8668	8412	-2,95%
<i>car6.csv</i>	J8M9	9359	9101	-2,75%
<i>car7.csv</i>	J7M7	7368	7187	-2,45%
<i>car8.csv</i>	J8M8	9386	8879	-5,40%
<i>rec01.csv</i>	J20M5	1672	1629	-3,49%
<i>rec03.csv</i>	J20M5	1564	1446	-7,54%
<i>rec05.csv</i>	J20M5	1617	1517	-5,24%
<i>rec07.csv</i>	J20M10	2100	2008	-4,38%
<i>rec09.csv</i>	J20M10	1948	1922	-1,33%
<i>rec11.csv</i>	J20M10	2018	1842	-8,72%
<i>rec13.csv</i>	J20M15	2552	2409	-5,60%
<i>rec15.csv</i>	J20M15	2505	2453	-2,07%
<i>rec17.csv</i>	J20M15	2392	2343	-2,04%
<i>rec19.csv</i>	J30M10	3227	3062	-5,11%
<i>rec21.csv</i>	J30M10	2931	2787	-4,91%
<i>rec23.csv</i>	J30M10	3008	2955	-1,76%
<i>rec25.csv</i>	J30M15	3711	3483	-6,14%
<i>rec27.csv</i>	J30M15	3551	3374	-4,98%
<i>rec29.csv</i>	J30M15	3391	3345	-1,35%
<i>rec31.csv</i>	J50M10	5042	4860	-3,60%
<i>rec33.csv</i>	J50M10	4869	4773	-1,97%
<i>rec35.csv</i>	J50M10	5236	4995	-4,60%
<i>rec37.csv</i>	J75M20	8398	8045	-4,20%
<i>rec39.csv</i>	J75M20	9486	8225	-13,29%
<i>rec41.csv</i>	J75M20	9142	8933	-2,28%

Pada Tabel 5.21 dan Tabel 5.22 ditunjukkan bahwa metode GA-TS dapat mengungguli metode heuristik GA. Metode TS

dapat membantu untuk mendapatkan nilai *makespan* yang lebih baik dan dapat meningkatkan kinerja dari metode GA dengan penurunan rata-rata hasil 4,45%.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang dapat diambil dari hasil uji coba dan evaluasi yang telah dilakukan. Selain kesimpulan, terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak nantinya.

6.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil uji coba dan evaluasi pada penyelesaian permasalahan NWFS dengan menggunakan metode GA-TS adalah sebagai berikut:

1. Metode GA-TS dapat menyelesaikan permasalahan NWFS dengan mendapatkan nilai *makespan* yang minimal.
2. Metode TS dapat meningkatkan kinerja dari metode GA dengan penurunan rata-rata hasil sebesar 4,45%.
3. Secara keseluruhan metode GA-TS dapat mengungguli kinerja dari dua metode lainnya, yaitu metode dari penyedia data uji coba Carlier dan Reeves dengan keunggulan 93,10% dan metode Algoritma Genetika dari Chaudry & Khan dengan keunggulan 68,96%.
4. Parameter-parameter yang terdapat pada metode GA-TS memegang peranan penting dalam penyelesaian permasalahan NWFS. Parameter-parameter tersebut adalah
 - a. Parameter jumlah populasi dengan 70 populasi.
 - b. Parameter nilai awal iterasi *Tabu Search* dengan nilai 25.
 - c. Parameter batas iterasi *Tabu Search* dengan nilai 0,1.
 - d. Parameter kawin silang dengan nilai 0,85.
 - e. Parameter mutasi dengan nilai 0,001.

6.2 Saran

Saran yang diberikan terkait pengerjaan penelitian ini adalah dilakukan uji coba dengan nilai yang lebih bervariasi pada parameter-parameter sebagai berikut:

- a. Parameter jumlah populasi
- b. Parameter kawin silang
- c. Parameter mutasi
- d. Parameter nilai awas iterasi *Tabu Search*
- e. Parameter batas iterasi *Tabu Search*

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan yang dapat diambil dari hasil uji coba dan evaluasi yang telah dilakukan. Selain kesimpulan, terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak nantinya.

6.1 Kesimpulan

Kesimpulan yang didapatkan berdasarkan hasil uji coba dan evaluasi pada penyelesaian permasalahan NWFS dengan menggunakan metode GA-TS adalah sebagai berikut:

1. Metode GA-TS dapat menyelesaikan permasalahan NWFS dengan mendapatkan nilai *makespan* yang minimal.
2. Metode TS dapat meningkatkan kinerja dari metode GA dengan penurunan rata-rata hasil sebesar 4,45%.
3. Secara keseluruhan metode GA-TS dapat mengungguli kinerja dari dua metode lainnya, yaitu metode dari penyedia data uji coba Carlier dan Reeves dengan keunggulan 93,10% dan metode Algoritma Genetika dari Chaudry & Khan dengan keunggulan 68,96%.
4. Parameter-parameter yang terdapat pada metode GA-TS memegang peranan penting dalam penyelesaian permasalahan NWFS. Parameter-parameter tersebut adalah
 - a. Parameter jumlah populasi dengan 70 populasi.
 - b. Parameter nilai awal iterasi *Tabu Search* dengan nilai 25.
 - c. Parameter batas iterasi *Tabu Search* dengan nilai 0,1.
 - d. Parameter kawin silang dengan nilai 0,85.
 - e. Parameter mutasi dengan nilai 0,001.

6.2 Saran

Saran yang diberikan terkait pengerjaan penelitian ini adalah dilakukan uji coba dengan nilai yang lebih bervariasi pada parameter-parameter sebagai berikut:

- a. Parameter jumlah populasi
- b. Parameter kawin silang
- c. Parameter mutasi
- d. Parameter nilai awas iterasi *Tabu Search*
- e. Parameter batas iterasi *Tabu Search*

DAFTAR PUSTAKA

- [1] C. Blum und A. Roli, „Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison,“ *ACM Computing Surveys (CSUR)*, Bd. 35, Nr. 3, pp. 268-308, 2003.
- [2] F. Glover und M. Laguna, *Tabu Search*, Boston: Kluwer Academic Publishers, 1997.
- [3] J. Carlier, „Ordonnancements a Contraintes Disjonctives,“ *Rairo-Rech Oper.*, Bd. 12, Nr. 4, pp. 333-351, 1978.
- [4] C. R. Reeves, „A Genetic Algorithm For Flowshop Sequencing,“ *Comput. Oper. Res.*, Bd. 22, Nr. 1, pp. 5-13, 1995.
- [5] I. A. Chaudry und A. M. Khan, „Minimizing Makespan For a No-Wait Flowshop Using Genetic Algorithm,“ *Indian Academy of Sciences*, pp. 695-707, 2012.
- [6] S. S. Reddi und C. V. Ramamoorthy, „On the Flowshop Sequencing Problem With No-wait in Process.,“ *Operational Research*, Bd. 23, Nr. 3, pp. 323-331, 1972.
- [7] T. W. Gwiazda, *Genetic Algorithms Reference Vol.1 Crossover for single-objective numerical optimization problems*, Lomianski, 2006.
- [8] L. Wen-Yang, L. Wen-Yuan und H. Tzung-Pei, „Adapting Crossover and Mutation Rates in Genetic Algorithms,“ *Journal of Information Science and Engineering*, Bd. 19, pp. 889-903, 2003.

LAMPIRAN A IMPLEMENTASI PROGRAM UTAMA

1	%--Data
2	data = xlsread('J7M5.csv');
3	mesin = size (data, 2);
4	job = size(data, 1);
5	populasi = 70;
6	%--Parameter
7	m = 0.005;
8	cr= 0.85;
9	parTS = round(0.1*populasi);
10	iterasiTS = 25;
11	batasIterasiTS = 0.1;
12	alphaTS = 0.7;
13	%--Membangkitkan populasi
14	for i = 1 : populasi
15	temp = randperm(job);
16	for j = 1 : job
17	kromosom (i, j) = temp (1, j);
18	end
19	end
20	%--Menghitung makespan
21	timeKromosom(populasi)=0;
22	for i=1:populasi
23	[waktu]= fitness2(data,kromosom(i, :),mesin,job);
24	timeKromosom(i)=waktu;
25	end
26	%--Tranpose makespan
27	timeKromosom2 = transpose (timeKromosom) ;
28	[waktuNWFs urutan] = sort (timeKromosom2)
29	%--Mengurutkan kromosom sesuai makespan
30	for i = 1 : populasi
31	parent(i,:) = kromosom(urutan(i, 1),:);
32	end

Kode Sumber Lampiran A1

33	[waktuNWFS urutan] = sort (waktuNWFS);
34	<i>%--Metode Tabu Search</i>
35	kromosomTS = zeros (parTS*3, size(kromosom,2));
36	<i>%--Makespan Kecil</i>
37	for i = 1 : parTS
38	kromosomTS(i,:) = parent(i,:)
39	waktuTS(i) = waktuNWFS(i,1)
40	end
41	<i>%--Makespan Sedang</i>
42	middleTS = round(populasi/2) - floor(parTS/2);
43	for i = parTS+1 : parTS*2
44	kromosomTS(i,:) = parent(middleTS,:)
45	waktuTS(i) = waktuNWFS(middleTS,1)
46	middleTS = middleTS + 1;
47	end
48	<i>%--Makespan Besar</i>
49	downTS = populasi+1 - parTS;
50	for i = parTS*2 + 1 : parTS*3
51	kromosomTS(i,:) = parent(downTS,:)
52	waktuTS(i) = waktuNWFS(downTS,1)
53	downTS = downTS + 1;
54	end
55	<i>%--Proses TS</i>
56	ii = 1;
57	while (iterasiTS > batasIterasiTS)
58	<i>%--Mutasi TS</i>
59	jumlahMutasi = size(kromosomTS, 1);
60	for i=1:jumlahMutasi
61	posisi = randperm(job,2);
62	p1 = min (posisi);
63	p2 = max (posisi);
64	tempKromosomTS(i,:) = kromosomTS(i,:);
65	mutasiTemp = tempKromosomTS(i,p1);
66	tempKromosomTS(i,p1) = tempKromosomTS(i,p2);

Kode Sumber Lampiran A2

67	tempKromosomTS(i,p2) = mutasiTemp;
68	<i>%--Perhitungan Nilai Makespan</i>
69	[waktuMutasi(i)] = fitness2(data,tempKromosomTS(i,:),mesin,job
70	if waktuMutasi(1, i) <= waktuTS(1,i)
71	<i>%--lebih kecil sama dengan</i>
72	kromosomTS(i,:) = tempKromosomTS(i,:)
73	else
74	<i>%--Probabilitas Menjadi</i>
75	TS = exp((- (waktuMutasi(1,i) - waktuTS(1,i)))/iterasiTS)
76	y = rand;
77	if TS >= y
78	kromosomTS(i,:) = tempKromosomTS(i,:)
79	end
80	end
81	end
82	iterasiTS = iterasiTS * alphaTS ^ ii;
83	ii = ii + 1
84	end
85	<i>%--Penggabungan Kromosom Hasil TS ke</i>
86	cc = populasi+1;
87	waktuMutasi1 = transpose (waktuMutasi);
88	for i = 1 : size(kromosomTS, 1)
89	parent (cc, :) = kromosomTS(i,:);
90	waktuNWFS (cc, 1) = waktuMutasi1(i, 1);
91	cc = cc + 1;
92	end
93	<i>%--Mengurutkan Makespan Ascending</i>
94	[waktuNWFS urutan] = sort (waktuNWFS);
95	<i>%--Mengurutkan Kromosom Sesuai Makespan</i>
96	for i = 1 : populasi
97	parent(i,:) = parent(urutan(i,1),:);
98	end

Kode Sumber Lampiran A3

99	%--Mencari kromosom untuk di Crossover
100	for i = 1 : populasi
101	r1(i) = rand;
102	end
103	availCR = find (r1 < cr);
104	iterCR = size(availCR,2);
105	%--Proses crossover
106	for i = 1 : iterCR
107	if i == populasi
108	[child1 child2] = crossover(parent(i,:), parent(1,:));
109	else
110	[child1 child2] = crossover(parent(i,:), parent(i+1,:));
111	end
112	%--Hitung fitness setiap child
113	[waktuChild1] = fitness2(data,child1,mesin,job);
114	[waktuChild2] = fitness2(data,child2,mesin,job);
115	%--Seleksi child yang terbaik
116	if waktuChild1 < waktuChild2
117	child(i,:) = child1;
118	else
119	child(i,:) = child2;
120	end
121	end
122	%--Mencari Child untuk di Mutasi
123	for i=1 : size(child,1)
124	r2(i) = rand;
125	end
126	availM = find (r2 < m);
127	iterM = size(availM,2)-1;
128	%--Proses Mutasi
129	for i = 1 : iterM
130	posisi = randperm(job,2);
131	p1 = min (posisi);
132	p2 = max (posisi);

Kode Sumber Lampiran A4

133	mutasiTemp = child (availM(i),p1);
134	child (availM(i),p1) = child (availM(i),p2);
135	child (availM(i),p2) = mutasiTemp;
136	end
137	%--Menggabungkan populasi awal dengan
138	merge = [parent;child];
139	%--Hitung Makespan kromosom Hasil gabungan
140	waktuKromosomFinal (size (parent,1))=0;
141	for i=1:size (merge,1)
142	[waktu]= fitness2 (data,merge (i,:),mesin,job);
143	waktuKromosomFinal (i)=waktu;
144	end
145	%--Transpose dan Sorting
146	waktuKromosomFinal2 = transpose (waktuKromosomFinal);
147	[bestWaktu urutan] = sort (waktuKromosomFinal2);
148	%--Mengurutkan kromosom sesuai Makespan
149	for q = 1 : size (merge,1)
150	bestKromosom (q,:) = merge (urutan (q,1) , :);
151	end
152	%--Display Solusi
153	disp ('Best Sequence No-Wait Flow-Shop')
154	disp (bestKromosom (1, :))
155	disp ('Makespan No-Wait Flow-Shop')
156	disp (bestWaktu (1,1))

Kode Sumber Lampiran A5

(Halaman ini sengaja dikosongkan)

LAMPIRAN B HASIL UJI KINERJA

Tabel Lampiran B1 Hasil Uji Coba Pada Data Uji *car1.csv*

<i>car1.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	8564	13	8460
2	8826	14	8724
3	8519	15	8343
4	8286	16	8481
5	8595	17	8731
6	8365	18	8402
7	8744	19	8460
8	8586	20	8724
9	8798	21	8343
10	7998	22	8481
11	8731	23	8601
12	8402	24	8356

Tabel Lampiran B2 Hasil Uji Coba Pada Data Uji *car2.csv*

<i>car2.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	9743	13	9845
2	9480	14	9513
3	9361	15	9240
4	9812	16	9549
5	8800	17	9793
6	9048	18	8758
7	9643	19	9823
8	9248	20	9512
9	9937	21	9756
10	9281	22	9496
11	9569	23	9677
12	9367	24	9134

Tabel Lampiran B3 Hasil Uji Coba Pada Data Uji car3.csv

<i>car3.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	9399	13	9364
2	8869	14	9481
3	9726	15	9481
4	9635	16	9438
5	9382	17	9690
6	9239	18	9858
7	10000	19	9795
8	9800	20	9233
9	9473	21	9040
10	8760	22	9712
11	8855	23	9656
12	9593	24	9599

Tabel Lampiran B4 Hasil Uji Coba Pada Data Uji car4.csv

<i>car4.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	10709	13	10144
2	10574	14	10652
3	10362	15	10781
4	10036	16	9979
5	10599	17	10688
6	10925	18	10279
7	11007	19	10396
8	10691	20	10824
9	10930	21	10887
10	10325	22	10694
11	10584	23	10642
12	10545	24	10299

Tabel Lampiran B5 Hasil Uji Coba Pada Data Uji car5.csv

<i>car5.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	8893	13	9366
2	8995	14	8523
3	9293	15	8760
4	8685	16	9169
5	8523	17	8412
6	8794	18	8608
7	9139	19	9132
8	9240	20	8933
9	8964	21	8908
10	9146	22	9096
11	9239	23	9079
12	8997	24	8651

Tabel Lampiran B6 Hasil Uji Coba Pada Data Uji car6.csv

<i>car6.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	9101	13	9821
2	9369	14	9572
3	9752	15	9519
4	9790	16	9714
5	9708	17	9646
6	9426	18	9735
7	9295	19	9642
8	9181	20	9526
9	9566	21	9526
10	9659	22	9526
11	9490	23	9677
12	9847	24	9896

Tabel Lampiran B7 Hasil Uji Coba Pada Data Uji car7.csv

<i>car7.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	7369	13	7187
2	7368	14	7369
3	7187	15	7369
4	7369	16	7369
5	7204	17	7260
6	7204	18	7187
7	7369	19	7374
8	7369	20	7587
9	7369	21	7374
10	7369	22	7280
11	7369	23	7467
12	7368	24	7502

Tabel Lampiran B8 Hasil Uji Coba Pada Data Uji car8.csv

<i>car8.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	8879	13	9169
2	9429	14	9147
3	9270	15	9087
4	9077	16	9311
5	9319	17	9274
6	9063	18	9378
7	9335	19	9090
8	9076	20	8982
9	9077	21	9054
10	9246	22	9161
11	9053	23	9146
12	9349	24	9162

Tabel Lampiran B9 Hasil Uji Coba Pada Data Uji rec01.csv

<i>rec01.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	1778	13	1758
2	1741	14	1743
3	1728	15	1737
4	1750	16	1767
5	1754	17	1714
6	1762	18	1771
7	1665	19	1629
8	1797	20	1773
9	1766	21	1746
10	1773	22	1722
11	1715	23	1701
12	1730	24	1682

Tabel Lampiran B10 Hasil Uji Coba Pada Data Uji rec03.csv

<i>rec03.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	1563	13	1629
2	1554	14	1568
3	1593	15	1571
4	1593	16	1566
5	1573	17	1526
6	1593	18	1623
7	1593	19	1591
8	1446	20	1574
9	1603	21	1654
10	1626	22	1633
11	1596	23	1544
12	1569	24	1550

Tabel Lampiran B11 Hasil Uji Coba Pada Data Uji rec05.csv

<i>rec05.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	1540	13	1631
2	1577	14	1618
3	1553	15	1621
4	1594	16	1635
5	1644	17	1597
6	1656	18	1576
7	1542	19	1517
8	1535	20	1634
9	1594	21	1608
10	1594	22	1571
11	1594	23	1645
12	1609	24	1601

Tabel Lampiran B12 Hasil Uji Coba Pada Data Uji rec07.csv

<i>rec07.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2124	13	2215
2	2193	14	2197
3	2146	15	2175
4	2211	16	2215
5	2170	17	2170
6	2220	18	2008
7	2046	19	2217
8	2127	20	2187
9	2306	21	2226
10	2212	22	2124
11	2131	23	2190
12	2096	24	2190

Tabel Lampiran B13 Hasil Uji Coba Pada Data Uji rec09.csv

<i>rec09.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2052	13	2085
2	2067	14	2009
3	2095	15	1922
4	2022	16	2104
5	2071	17	2007
6	1976	18	2097
7	2116	19	2023
8	2066	20	2069
9	2035	21	2093
10	1953	22	2082
11	2075	23	1994
12	2048	24	2053

Tabel Lampiran B14 Hasil Uji Coba Pada Data Uji rec11.csv

<i>rec11.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2049	13	1842
2	1995	14	1954
3	1943	15	2053
4	1934	16	1970
5	1963	17	1874
6	1990	18	2030
7	2007	19	2065
8	1976	20	2030
9	1986	21	2040
10	2013	22	2016
11	2058	23	1970
12	2003	24	1889

Tabel Lampiran B15 Hasil Uji Coba Pada Data Uji *rec13.csv*

<i>rec13.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2575	13	2647
2	2497	14	2566
3	2613	15	2591
4	2543	16	2607
5	2573	17	2535
6	2665	18	2600
7	2604	19	2642
8	2602	20	2465
9	2409	21	2636
10	2533	22	2582
11	2657	23	2592
12	2642	24	2597

Tabel Lampiran B16 Hasil Uji Coba Pada Data Uji *rec15.csv*

<i>rec15.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2566	13	2572
2	2629	14	2599
3	2621	15	2536
4	2543	16	2547
5	2527	17	2626
6	2540	18	2501
7	2567	19	2585
8	2578	20	2453
9	2499	21	2649
10	2652	22	2519
11	2542	23	2522
12	2605	24	2671

Tabel Lampiran B17 Hasil Uji Coba Pada Data Uji rec17.csv

<i>rec17.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2343	13	2449
2	2509	14	2464
3	2450	15	2519
4	2525	16	2438
5	2478	17	2453
6	2444	18	2476
7	2471	19	2490
8	2468	20	2432
9	2409	21	2551
10	2550	22	2358
11	2459	23	2472
12	2470	24	2511

Tabel Lampiran B18 Hasil Uji Coba Pada Data Uji rec19.csv

<i>rec19.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	3228	13	3200
2	3240	14	3314
3	3266	15	3277
4	3270	16	3115
5	3168	17	3082
6	3217	18	3235
7	3278	19	3128
8	3314	20	3062
9	3273	21	3124
10	3257	22	3297
11	3273	23	3109
12	3247	24	3252

Tabel Lampiran B19 Hasil Uji Coba Pada Data Uji rec21.csv

<i>rec21.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	2939	13	3102
2	3104	14	3028
3	3094	15	3058
4	2999	16	3027
5	2977	17	2938
6	2995	18	2985
7	3125	19	2908
8	3048	20	2991
9	2973	21	2918
10	2884	22	2787
11	3080	23	2843
12	3021	24	3016

Tabel Lampiran B20 Hasil Uji Coba Pada Data Uji rec23.csv

<i>rec23.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	3118	13	2992
2	3256	14	2992
3	2995	15	2992
4	3184	16	3073
5	3100	17	3156
6	3142	18	3243
7	3104	19	3033
8	3103	20	3071
9	3136	21	3112
10	3189	22	3006
11	2955	23	3112
12	3101	24	3139

Tabel Lampiran B21 Hasil Uji Coba Pada Data Uji rec25.csv

<i>rec25.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	3699	13	3759
2	3804	14	3836
3	3714	15	3802
4	3483	16	3727
5	3751	17	3818
6	3763	18	3867
7	3738	19	3883
8	3811	20	3780
9	3743	21	3770
10	3598	22	3813
11	3912	23	3721
12	3808	24	3894

Tabel Lampiran B22 Hasil Uji Coba Pada Data Uji rec27.csv

<i>rec27.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	3665	13	3716
2	3746	14	3695
3	3682	15	3583
4	3693	16	3374
5	3587	17	3761
6	3607	18	3775
7	3646	19	3617
8	3646	20	3636
9	3643	21	3777
10	3675	22	3675
11	3681	23	3743
12	3507	24	3623

Tabel Lampiran B23 Hasil Uji Coba Pada Data Uji rec29.csv

<i>rec29.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	3541	13	3623
2	3400	14	3519
3	3345	15	3600
4	3469	16	3598
5	3373	17	3607
6	3520	18	3619
7	3610	19	3626
8	3403	20	3499
9	3411	21	3668
10	3345	22	3680
11	3815	23	3367
12	3737	24	3402

Tabel Lampiran B24 Hasil Uji Coba Pada Data Uji rec31.csv

<i>rec31cs.v</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	5181	13	5037
2	5028	14	5253
3	5221	15	5039
4	5001	16	5106
5	4975	17	4987
6	5091	18	5038
7	5041	19	5114
8	5032	20	4928
9	5150	21	5111
10	5075	22	5220
11	5043	23	5118
12	5244	24	4860

Tabel Lampiran B25 Hasil Uji Coba Pada Data Uji rec33.csv

<i>rec33.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	4978	13	5007
2	5084	14	4946
3	4990	15	4998
4	5073	16	5078
5	4906	17	5061
6	5166	18	4973
7	4976	19	4847
8	5128	20	4991
9	5113	21	4773
10	5046	22	5082
11	5049	23	5098
12	5139	24	4920

Tabel Lampiran B26 Hasil Uji Coba Pada Data Uji rec35.csv

<i>rec35csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	5188	13	5509
2	5265	14	5349
3	5225	15	5305
4	5427	16	5420
5	5413	17	5339
6	5213	18	5234
7	5329	19	5344
8	5271	20	5226
9	5322	21	5164
10	5374	22	4995
11	5331	23	5307
12	5319	24	5343

Tabel Lampiran B27 Hasil Uji Coba Pada Data Uji rec37.csv

<i>rec37.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	8994	13	8735
2	8702	14	8934
3	8620	15	8975
4	9078	16	9026
5	9064	17	8936
6	8045	18	8828
7	8612	19	9008
8	8986	20	9001
9	9100	21	9037
10	8893	22	8826
11	8917	23	9130
12	8917	24	9142

Tabel Lampiran B28 Hasil Uji Coba Pada Data Uji rec39.csv

<i>rec39.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	9516	13	9469
2	9485	14	9749
3	9506	15	9850
4	9569	16	9675
5	9285	17	9522
6	9534	18	9212
7	9534	19	9686
8	9935	20	9660
9	9653	21	9603
10	9561	22	9433
11	9717	23	9395
12	9743	24	9789

Tabel Lampiran B29 Hasil Uji Coba Pada Data Uji rec41.csv

<i>rec41.csv</i>			
Iterasi ke-	<i>Makespan</i>	Iterasi ke-	<i>Makespan</i>
1	9281	13	9549
2	9367	14	9512
3	9489	15	9350
4	9408	16	9518
5	8990	17	9088
6	9217	18	9411
7	9377	19	9455
8	9086	20	9204
9	9377	21	9411
10	9531	22	9315
11	8933	23	9408
12	9527	24	9372

(Halaman ini sengaja dikosongkan)

LAMPIRAN C

HASIL UJI PERBANDINGAN PARAMETER

**Tabel Lampiran C1 Uji Perbandingan Parameter Jumlah
Populasi Pada *car1.csv***

Iterasi Ke-	car1.csv			Iterasi Ke-	car1.csv		
	25	70	100		25	70	100
1	8648	8564	8059	13	8648	8460	8469
2	8479	8826	8679	14	8517	8724	8461
3	8648	8519	8338	15	8636	8343	8673
4	8648	8286	8398	16	8648	8481	8296
5	8648	8595	8629	17	8648	8731	8050
6	8648	8365	8263	18	8648	8402	8613
7	8611	8744	8702	19	8610	8460	8050
8	8648	8586	8418	20	8648	8724	8581
9	8648	8798	8734	21	8460	8343	8449
10	8388	7998	8332	22	8235	8481	9088
11	8496	8731	8420	23	8648	8601	8512
12	8648	8402	8819	24	8648	8356	8569

**Tabel Lampiran C2 Uji Perbandingan Parameter Jumlah
Populasi Pada *car5.csv***

Iterasi Ke-	car1.csv			Iterasi Ke-	car1.csv		
	25	70	100		25	70	100
1	9127	8893	9056	13	9079	9366	8882
2	9504	8995	9234	14	8619	8523	8882
3	9497	9293	9093	15	8619	8760	8947
4	9165	8685	8476	16	8619	9169	9020
5	9401	8523	8812	17	8880	8412	8921
6	8996	8794	8848	18	9524	8608	9199
7	9338	9139	8890	19	9230	9132	9299
8	9082	9240	9423	20	9528	8933	9364
9	9421	8964	9117	21	9016	8908	9121
10	9585	9146	9192	22	9112	9096	9191
11	9678	9239	8882	23	9713	9079	9019
12	9179	8997	8882	24	9167	8651	9031

**Tabel Lampiran C3 Uji Perbandingan Parameter Jumlah
Populasi Pada *rec01.csv***

Iterasi Ke-	rec01.csv - Jumlah Populasi			Iterasi Ke-	rec01.csv - Jumlah Populasi		
	25	70	100		25	70	100
1	1834	1778	1709	13	1780	1758	1743
2	1677	1741	1772	14	1774	1743	1705
3	1759	1728	1681	15	1849	1737	1742
4	1740	1750	1717	16	1746	1767	1746
5	1733	1754	1719	17	1827	1714	1754
6	1678	1762	1703	18	1773	1771	1734
7	1764	1665	1742	19	1747	1629	1781
8	1781	1797	1661	20	1789	1773	1655
9	1749	1766	1760	21	1708	1746	1679
10	1821	1773	1731	22	1871	1722	1720
11	1784	1715	1718	23	1750	1701	1747
12	1682	1730	1742	24	1758	1682	1708

**Tabel Lampiran C4 Uji Perbandingan Parameter Jumlah
Populasi Pada *rec13.csv***

Iterasi Ke-	rec13.csv - Jumlah Populasi			Iterasi Ke-	rec13.csv - Jumlah Populasi		
	25	70	100		25	70	100
1	2616	2575	2608	13	2581	2647	252
2	2602	2497	2482	14	2510	2566	260
3	2615	2613	2562	15	2487	2591	257
4	2661	2543	2579	16	2686	2607	257
5	2626	2573	2594	17	2660	2535	263
6	2701	2665	2596	18	2572	2600	258
7	2701	2604	2532	19	2754	2642	267
8	2611	2602	2585	20	2674	2465	265
9	2701	2409	2575	21	2687	2636	261
10	2612	2533	2671	22	2678	2582	258
11	2681	2657	2681	23	2668	2592	250
12	2591	2642	2542	24	2711	2597	255

**Tabel Lampiran C5 Uji Perbandingan Parameter Jumlah
Populasi Pada *rec25.csv***

Iterasi Ke-	rec25.csv - Jumlah Populasi			Iterasi Ke-	rec25.csv - Jumlah Populasi		
	25	70	100		25	70	100
1	3846	3699	3760	13	3973	3759	3724
2	3809	3804	3803	14	3901	3836	3729
3	3787	3714	3814	15	4052	3802	3745
4	3861	3483	3856	16	3960	3727	3729
5	3861	3751	3786	17	3699	3818	3641
6	3861	3763	3779	18	3833	3867	3698
7	3843	3738	3821	19	3906	3883	3721
8	3964	3811	3827	20	4104	3780	3751
9	3888	3743	3712	21	3990	3770	3670
10	4004	3598	3849	22	3935	3813	3619
11	3803	3912	3687	23	3992	3721	3739
12	3887	3808	3757	24	3926	3894	3824

**Tabel Lampiran C6 Uji Perbandingan Parameter Jumlah
Populasi Pada *rec37.csv***

Iterasi Ke-	rec37.csv - Jumlah Populasi			Iterasi Ke-	rec37.csv – Jumlah Populasi		
	25	70	100		25	70	100
1	8892	8994	8782	13	8509	8735	8739
2	8781	8702	8752	14	8918	8934	8722
3	8688	8620	8752	15	8386	8975	8684
4	8718	9078	8617	16	8296	9026	8867
5	8718	9064	8051	17	8643	8936	8841
6	8610	8045	8522	18	881	8828	8841
7	8729	8612	8455	19	8908	9008	8529
8	8981	8986	8541	20	8866	9001	8613
9	8599	9100	8810	21	8553	9037	8848
10	8685	8893	8513	22	8556	8826	8531
11	8635	8917	8321	23	8767	9130	8631
12	8751	8917	8051	24	8663	9142	8988

**Tabel Lampiran C7 Uji Perbandingan Parameter Nilai Awal
Iterasi TS Pada *car1.csv***

Iterasi Ke-	car1.csv - Nilai Awal Iterasi TS			Iterasi Ke-	car1.csv - Nilai Awal Iterasi TS		
	10	25	50		10	25	50
1	8849	8564	8571	13	8756	8460	8678
2	8944	8826	8898	14	8067	8724	8843
3	8523	8519	8280	15	8466	8343	8177
4	8734	8286	8384	16	8576	8481	8641
5	8416	8595	8414	17	8492	8731	8469
6	8416	8365	9057	18	8793	8402	8780
7	8841	8744	8394	19	8662	8460	8468
8	8876	8586	8693	20	8939	8724	8881
9	8894	8798	8866	21	9133	8343	8643
10	8623	7998	8707	22	8775	8481	8720
11	8361	8731	8707	23	8899	8601	8425
12	8692	8402	8746	24	8613	8356	8126

**Tabel Lampiran C8 Uji Perbandingan Parameter Nilai Awal
Iterasi TS Pada *car5.csv***

Iterasi Ke-	car5.csv -Nilai Awal Iterasi TS			Iterasi Ke-	car5.csv -Nilai Awal Iterasi TS		
	10	25	50		10	25	50
1	9392	8893	9210	13	8940	9366	8719
2	9152	8995	8753	14	8979	8523	9099
3	9279	9293	8412	15	9388	8760	9151
4	8812	8685	9204	16	8665	9169	8508
5	9162	8523	9076	17	9119	8412	8604
6	8771	8794	8934	18	8368	8608	8770
7	9139	9139	8734	19	9004	9132	8310
8	9125	9240	9333	20	9241	8933	8794
9	9061	8964	8436	21	8482	8908	9085
10	8949	9146	8658	22	9143	9096	9462
11	8914	9239	9231	23	9493	9079	9409
12	8867	8997	9254	24	8766	8651	9046

Tabel Lampiran C9 Uji Perbandingan Parameter Nilai Awal Iterasi TS Pada *rec01.csv*

Iterasi Ke-	rec01.csv - Nilai Awal Iterasi TS			Iterasi Ke-	rec01.csv - Nilai Awal Iterasi TS		
	10	25	50		10	25	50
1	1776	1778	1704	13	170	1758	1690
2	1724	1741	1746	14	172	1743	1771
3	1769	1728	1725	15	176	1737	1746
4	1738	1750	1775	16	172	1767	1749
5	1770	1754	1760	17	164	1714	1750
6	1736	1762	1784	18	173	1771	1794
7	1696	1665	1726	19	173	1629	1727
8	1711	1797	1785	20	173	1773	1757
9	1743	1766	1778	21	173	1746	1756
10	1722	1773	1743	22	173	1722	1655
11	1726	1715	1703	23	173	1701	1737
12	1748	1730	1749	24	170	1682	1785

Tabel Lampiran C10 Uji Perbandingan Parameter Nilai Awal Iterasi TS Pada *rec13.csv*

Iterasi Ke-	rec13.csv - Nilai Awal Iterasi TS			Iterasi Ke-	rec13.csv - Nilai Awal Iterasi TS		
	10	25	50		10	25	50
1	2694	2575	2572	13	2503	2647	2565
2	2614	2497	2627	14	2526	2566	2607
3	2607	2613	2459	15	2614	2591	2656
4	2614	2543	2437	16	2492	2607	2618
5	2596	2573	2650	17	2509	2535	2528
6	2599	2665	2605	18	2454	2600	2644
7	2540	2604	2586	19	2622	2642	2527
8	2511	2602	2605	20	2606	2465	2647
9	2559	2409	2605	21	2525	2636	2578
10	2614	2533	2487	22	2640	2582	2408
11	2614	2657	2605	23	2520	2592	2521
12	2614	2642	2460	24	2505	2597	2573

Tabel Lampiran C11 Uji Perbandingan Parameter Nilai Awal Iterasi TS Pada *car1.csv*

Iterasi Ke-	rec25.csv -Nilai Awal Iterasi TS			Iterasi Ke-	rec25.csv - Nilai Awal Iterasi TS		
	10	25	50		10	25	50
1	3619	3699	3826	13	3904	3759	3859
2	3754	3804	3885	14	3799	3836	3887
3	3827	3714	3772	15	3702	3802	3753
4	3837	3483	3924	16	3749	3727	3805
5	3848	3751	3816	17	3763	3818	3812
6	3867	3763	3566	18	3868	3867	3820
7	3527	3738	3685	19	3830	3883	3792
8	3898	3811	3563	20	3859	3780	3717
9	3903	3743	3787	21	3669	3770	3768
10	3896	3598	3589	22	3768	3813	3749
11	3758	3912	3802	23	3775	3721	3635
12	3754	3808	3830	24	3950	3894	3824

Tabel Lampiran C12 Uji Perbandingan Parameter Nilai Awal Iterasi TS Pada *car1.csv*

Iterasi Ke-	rec37.csv - Nilai Awal Iterasi TS			Iterasi Ke-	rec37.csv - Nilai Awal Iterasi TS		
	10	25	50		10	25	50
1	8850	8994	8888	13	8932	8735	8875
2	8521	8702	8979	14	8726	8934	8941
3	8833	8620	8590	15	8489	8975	8858
4	8885	9078	8596	16	8431	9026	8591
5	8677	9064	8381	17	8598	8936	8594
6	8905	8045	8845	18	8344	8828	8832
7	8808	8612	8454	19	8739	9008	8845
8	8447	8986	8668	20	8632	9001	8762
9	8460	9100	8794	21	8688	9037	8915
10	8774	8893	8895	22	8645	8826	8697
11	8870	8917	8933	23	8747	9130	8912
12	8902	8917	8694	24	8678	9142	8694

**Tabel Lampiran C13 Uji Perbandingan Parameter Batas
Iterasi TS Pada *car1.csv***

Iterasi Ke-	car1.csv - Batas Iterasi TS			Iterasi Ke-	car1.csv - Batas Iterasi TS		
	0,1	0,5	1,0		0,1	0,5	1,0
1	8564	8995	8266	13	8460	8740	8915
2	8826	8505	8516	14	8724	8531	8784
3	8519	8721	8315	15	8343	8114	8612
4	8286	8747	8282	16	8481	8477	8769
5	8595	8464	8403	17	8731	8677	8726
6	8365	8675	8573	18	8402	8349	8832
7	8744	8511	8661	19	8460	8581	8460
8	8586	8613	8401	20	8724	8301	8730
9	8798	8732	8353	21	8343	8378	8473
10	7998	8693	8672	22	8481	8296	8665
11	8731	8637	8836	23	8601	8554	8775
12	8402	8563	8032	24	8356	8730	8697

**Tabel Lampiran C14 Uji Perbandingan Parameter Batas
Iterasi TS Pada *car5.csv***

Iterasi Ke-	car5.csv -Batas Iterasi TS			Iterasi Ke-	car5.csv -Batas Iterasi TS		
	0,1	0,5	1,0		0,1	0,5	1,0
1	8893	8904	9126	13	9366	9438	9239
2	8995	8489	9272	14	8523	9006	8722
3	9293	9248	9014	15	8760	8935	9518
4	8685	8499	8817	16	9169	9161	8902
5	8523	8489	9478	17	8412	9372	8878
6	8794	8979	8412	18	8608	9382	9346
7	9139	8866	9262	19	9132	8916	8787
8	9240	9125	9235	20	8933	9049	9499
9	8964	9236	9364	21	8908	9438	9294
10	9146	9015	9406	22	9096	9292	9225
11	9239	9141	9371	23	9079	9106	9490
12	8997	9176	9006	24	8651	9108	9312

**Tabel Lampiran C15 Uji Perbandingan Parameter Batas
Iterasi TS Pada *rec01.csv***

Iterasi Ke-	rec01.csv - Batas Iterasi TS			Iterasi Ke-	rec01.csv - Batas Iterasi TS		
	0,1	0,5	1,0		0,1	0,5	1,0
1	1778	1692	1760	13	1758	1732	1706
2	1741	1736	1717	14	1743	1761	1789
3	1728	1745	1752	15	1737	1764	1741
4	1750	1745	1701	16	1767	1740	1770
5	1754	1714	1764	17	1714	1739	1646
6	1762	1727	1733	18	1771	1692	1713
7	1665	1744	1775	19	1629	1764	1702
8	1797	1721	1725	20	1773	1684	1739
9	1766	1760	1757	21	1746	1679	1800
10	1773	1756	1715	22	1722	1728	1766
11	1715	1672	1714	23	1701	1752	1800
12	1730	1764	1654	24	1682	1739	1807

**Tabel Lampiran C16 Uji Perbandingan Parameter Batas
Iterasi TS Pada *rec13.csv***

Iterasi Ke-	rec13.csv -Batas Iterasi TS			Iterasi Ke-	rec13.csv -Batas Iterasi TS		
	0,1	0,5	1,0		0,1	0,5	1,0
1	2575	2571	2572	13	2647	2626	2504
2	2497	2653	2627	14	2566	2618	2601
3	2613	2591	2648	15	2591	2683	2582
4	2543	2673	2659	16	2607	2547	2601
5	2573	2544	2717	17	2535	2500	2581
6	2665	2535	2482	18	2600	2625	2612
7	2604	2551	2569	19	2642	2620	2633
8	2602	2649	2601	20	2465	2510	2482
9	2409	2558	2601	21	2636	2622	2650
10	2533	2607	2559	22	2582	2608	2563
11	2657	2625	2601	23	2592	2627	2537
12	2642	2545	2582	24	2597	2626	2603

**Tabel Lampiran C17 Uji Perbandingan Parameter Batas
Iterasi TS Pada *rec25.csv***

Iterasi Ke-	rec25.csv -Batas Iterasi TS			Iterasi Ke-	rec25.csv -Batas Iterasi TS		
	0,1	0,5	1,0		0,1	0,5	1,0
1	3699	3894	3908	13	3759	3796	3704
2	3804	3748	3788	14	3836	3730	3800
3	3714	3794	3729	15	3802	3796	3696
4	3483	3872	3859	16	3727	3769	3839
5	3751	3837	3801	17	3818	3874	3638
6	3763	3779	3981	18	3867	3793	3680
7	3738	3850	3882	19	3883	3761	3716
8	3811	3828	3820	20	3780	3890	3689
9	3743	3555	3787	21	3770	3715	3836
10	3598	3736	3716	22	3813	3730	3842
11	3912	3817	3670	23	3721	3843	3842
12	3808	3804	3817	24	3894	3775	3768

**Tabel Lampiran C18 Uji Perbandingan Parameter Batas
Iterasi TS Pada *rec37.csv***

Iterasi Ke-	rec37.csv - Batas Iterasi TS			Iterasi Ke-	rec37.csv - Batas Iterasi TS		
	0,1	0,5	1,0		0,1	0,5	1,0
1	8994	8854	8753	13	8735	8947	8752
2	8702	8620	8381	14	8934	8442	8432
3	8620	8739	8923	15	8975	8740	8931
4	9078	8858	8626	16	9026	8400	8828
5	9064	8736	8412	17	8936	8870	8740
6	8045	8767	8760	18	8828	8909	8812
7	8612	8796	8857	19	9008	8873	8767
8	8986	8820	8940	20	9001	8803	8745
9	9100	8822	8846	21	9037	8554	8563
10	8893	8664	8825	22	8826	8804	8284
11	8917	8863	8787	23	9130	8894	8695
12	8917	8925	8298	24	9142	8810	8781

**Tabel Lampiran C19 Uji Perbandingan Parameter Kawin
Silang Pada *car1.csv***

Iterasi Ke-	car1.csv - Kawin Silang			Iterasi Ke-	car1.csv - Kawin Silang		
	0,65	0,85	0,90		0,65	0,85	0,90
1	8535	8564	9129	13	8448	8460	8607
2	8855	8826	8609	14	8362	8724	8500
3	8285	8519	8651	15	8425	8343	8829
4	8556	8286	8771	16	8627	8481	8943
5	8772	8595	8692	17	8066	8731	8875
6	8481	8365	8616	18	8514	8402	8641
7	8188	8744	8668	19	8394	8460	8811
8	8635	8586	8341	20	8736	8724	8151
9	8268	8798	8238	21	8567	8343	8368
10	8554	7998	8902	22	8621	8481	8107
11	8863	8731	8553	23	8924	8601	8368
12	8563	8402	8552	24	8785	8356	8732

**Tabel Lampiran C20 Uji Perbandingan Parameter Kawin
Silang Pada *car5.csv***

Iterasi Ke-	car5.csv - Kawin Silang			Iterasi Ke-	car5.csv - Kawin Silang		
	0,65	0,85	0,90		0,65	0,85	0,90
1	8745	8893	9031	13	9187	9366	9099
2	9252	8995	9243	14	8628	8523	9175
3	8907	9293	8794	15	9059	8760	9425
4	9140	8685	8512	16	9085	9169	8873
5	9077	8523	8812	17	9704	8412	8901
6	8986	8794	9120	18	8943	8608	8900
7	9388	9139	9341	19	9134	9132	8859
8	9302	9240	8343	20	8812	8933	8966
9	8923	8964	9343	21	8628	8908	8869
10	9141	9146	8842	22	9247	9096	9058
11	8799	9239	8978	23	8812	9079	9115
12	9304	8997	9078	24	9463	8651	9087

**Tabel Lampiran C21 Uji Perbandingan Parameter Kawin
Silang Pada *rec01.csv***

Iterasi Ke-	rec01.csv - Kawin Silang			Iterasi Ke-	rec01.csv - Kawin Silang		
	0,65	0,85	0,90		0,65	0,85	0,90
1	1773	1778	1775	13	1721	1758	1757
2	1708	1741	1725	14	1785	1743	1729
3	1788	1728	1727	15	1713	1737	1749
4	1794	1750	1727	16	1750	1767	1758
5	1711	1754	1746	17	1776	1714	1715
6	1766	1762	1799	18	1802	1771	1765
7	1671	1665	1799	19	1743	1629	1734
8	1731	1797	1799	20	1776	1773	1714
9	1758	1766	1750	21	1812	1746	1790
10	1786	1773	1696	22	1776	1722	1803
11	1772	1715	1776	23	1718	1701	1708
12	1805	1730	1751	24	1748	1682	1631

**Tabel Lampiran C22 Uji Perbandingan Parameter Kawin
Silang Pada *rec13.csv***

Iterasi Ke-	rec13.csv - Kawin Silang			Iterasi Ke-	rec13.csv - Kawin Silang		
	0,65	0,85	0,90		0,65	0,85	0,90
1	2445	2575	2604	13	2656	2647	2643
2	2714	2497	2618	14	2604	2566	2430
3	2661	2613	2645	15	2679	2591	2545
4	2601	2543	2698	16	2658	2607	2492
5	2688	2573	2513	17	2661	2535	2551
6	2598	2665	2617	18	2459	2600	2625
7	2631	2604	2617	19	2665	2642	2436
8	2593	2602	2566	20	2607	2465	2536
9	2656	2409	2662	21	2562	2636	2613
10	2656	2533	2635	22	2600	2582	2537
11	2541	2657	2650	23	2653	2592	2553
12	2456	2642	2546	24	2571	2597	2678

**Tabel Lampiran C23 Uji Perbandingan Parameter Kawin
Silang Pada *rec25.csv***

Iterasi Ke-	rec25.csv -Kawin Silang			Iterasi Ke-	rec25.csv -Kawin Silang		
	0,65	0,85	0,90		0,65	0,85	0,90
1	3710	3699	3850	13	3735	3759	3830
2	3785	3804	3832	14	3927	3836	3820
3	3829	3714	3883	15	3788	3802	3633
4	3854	3483	3788	16	3851	3727	3685
5	3881	3751	3668	17	3694	3818	3933
6	3789	3763	3828	18	3738	3867	3900
7	3692	3738	3881	19	3837	3883	3827
8	3630	3811	3794	20	3756	3780	3706
9	3833	3743	3762	21	3835	3770	3711
10	3790	3598	3768	22	3843	3813	3744
11	3755	3912	3749	23	3899	3721	3773
12	3885	3808	3822	24	3851	3894	3848

**Tabel Lampiran C24 Uji Perbandingan Parameter Kawin
Silang Pada *rec37.csv***

Iterasi Ke-	rec37.csv - Kawin Silang			Iterasi Ke-	rec37.csv - Kawin Silang		
	0,65	0,85	0,90		0,65	0,85	0,90
1	8666	8994	8707	13	8786	8735	8822
2	8566	8702	8681	14	8621	8934	8841
3	8710	8620	8462	15	8899	8975	8509
4	8883	9078	8672	16	8647	9026	8862
5	8689	9064	8718	17	8839	8936	8758
6	8691	8045	8674	18	8729	8828	8197
7	8632	8612	8825	19	8542	9008	8556
8	8596	8986	8656	20	8843	9001	8981
9	8556	9100	8590	21	9001	9037	8631
10	8696	8893	8769	22	8742	8826	8846
11	8907	8917	8968	23	8718	9130	8754
12	8565	8917	8669	24	8726	9142	8788

**Tabel Lampiran C25 Uji Perbandingan Parameter Mutasi
Pada *car1.csv***

Iterasi Ke-	car1.csv - Mutasi			Iterasi Ke-	car1.csv - Mutasi		
	0,001	0,005	0,05		0,001	0,005	0,05
1	8564	8154	8676	13	8460	8253	8764
2	8826	8433	8621	14	8724	8453	8503
3	8519	8606	8163	15	8343	8629	8107
4	8286	8503	8314	16	8481	8629	8693
5	8595	8662	8267	17	8731	8387	8966
6	8365	8450	8267	18	8402	8791	8466
7	8744	8321	8494	19	8460	8513	8185
8	8586	8629	8268	20	8724	8671	8428
9	8798	8378	8764	21	8343	8408	8524
10	7998	8629	8485	22	8481	8786	8713
11	8731	8388	8764	23	8601	8449	8732
12	8402	8629	8500	24	8356	8510	8384

**Tabel Lampiran C26 Uji Perbandingan Parameter Mutasi
Pada *car5.csv***

Iterasi Ke-	car5.csv - Mutasi			Iterasi Ke-	car5.csv - Mutasi		
	0,00	0,005	0,05		0,001	0,005	0,05
1	8893	9086	9287	13	9366	9301	926
2	8995	9134	8734	14	8523	8927	909
3	9293	8956	8820	15	8760	8549	935
4	8685	8531	8993	16	9169	9138	911
5	8523	8531	8993	17	8412	9313	933
6	8794	8531	8466	18	8608	8454	915
7	9139	8531	8993	19	9132	9092	927
8	9240	9276	8993	20	8933	9120	882
9	8964	8493	8993	21	8908	9313	898
10	9146	8824	8993	22	9096	8987	912
11	9239	8807	9054	23	9079	9393	912
12	8997	9158	9571	24	8651	8838	943

**Tabel Lampiran C27 Uji Perbandingan Parameter Mutasi
Pada *rec01.csv***

Iterasi Ke-	rec01.csv - Mutasi			Iterasi Ke-	rec01.csv - Mutasi		
	0,001	0,005	0,05		0,001	0,005	0,05
1	1778	1646	1754	13	1758	1729	1737
2	1741	1703	1636	14	1743	1719	1693
3	1728	1722	1636	15	1737	1760	1775
4	1750	1731	1636	16	1767	1764	1739
5	1754	1748	1772	17	1714	1772	1784
6	1762	1697	1724	18	1771	1736	1665
7	1665	1809	1675	19	1629	1767	1769
8	1797	1752	1773	20	1773	1784	1772
9	1766	1723	1738	21	1746	1809	1779
10	1773	1698	1682	22	1722	1817	1767
11	1715	1750	1732	23	1701	1763	1693
12	1730	1764	1694	24	1682	1745	1781

**Tabel Lampiran C28 Uji Perbandingan Parameter Mutasi
Pada *rec13.csv***

Iterasi Ke-	rec13.csv - Mutasi			Iterasi Ke-	rec13.csv - Mutasi		
	0,001	0,005	0,05		0,001	0,005	0,05
1	2575	2663	2504	13	2647	2613	2622
2	2497	2593	2604	14	2566	2578	2564
3	2613	2758	2597	15	2591	2630	2497
4	2543	2633	2579	16	2607	2562	2628
5	2573	2588	2603	17	2535	2576	2654
6	2665	2613	2563	18	2600	2720	2409
7	2604	2553	2634	19	2642	2605	2593
8	2602	2587	2648	20	2465	2555	2616
9	2409	2668	2534	21	2636	2658	2606
10	2533	2587	2666	22	2582	2638	2591
11	2657	2588	2646	23	2592	2527	2610
12	2642	2505	2579	24	2597	2647	2539

**Tabel Lampiran C29 Uji Perbandingan Parameter Mutasi
Pada *rec25.csv***

Iterasi Ke-	rec25.csv - Mutasi			Iterasi Ke-	rec25.csv - Mutasi		
	0,001	0,005	0,05		0,001	0,005	0,05
1	3699	3828	3665	13	3759	3677	3644
2	3804	3831	3753	14	3836	3569	3813
3	3714	3733	3798	15	3802	3749	3726
4	3483	3728	3774	16	3727	3850	3686
5	3751	3739	3856	17	3818	3811	3896
6	3763	3795	3748	18	3867	3818	3689
7	3738	3794	3846	19	3883	3723	3774
8	3811	3875	3806	20	3780	3901	3851
9	3743	3737	3740	21	3770	3814	3662
10	3598	3856	3755	22	3813	3719	3703
11	3912	3862	3899	23	3721	3813	3768
12	3808	3787	3851	24	3894	3873	3746

**Tabel Lampiran C30 Uji Perbandingan Parameter Mutasi
Pada *rec37.csv***

Iterasi Ke-	rec37.csv - Mutasi			Iterasi Ke-	rec37.csv - Mutasi		
	0,001	0,005	0,05		0,001	0,005	0,05
1	8994	8782	8718	13	8735	8739	8848
2	8702	8752	8811	14	8934	8722	8944
3	8620	8752	8760	15	8975	8684	8795
4	9078	8617	8821	16	9026	8867	8694
5	9064	8051	8901	17	8936	8841	8804
6	8045	8522	8793	18	8828	8841	8689
7	8612	8455	8225	19	9008	8529	8867
8	8986	8541	8704	20	9001	8613	8727
9	9100	8810	8594	21	9037	8848	8308
10	8893	8513	8624	22	8826	8531	8555
11	8917	8321	8599	23	9130	8631	8559
12	8917	8051	8848	24	9142	8988	8400

(Halaman ini sengaja dikosongkan)

LAMPIRAN D

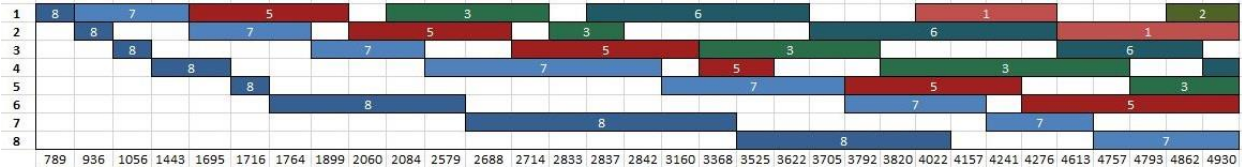
HASIL UJI VALIDITAS *GANTT CHART*

Tabel Lampiran D1 waktu proses *dataset car1.csv*

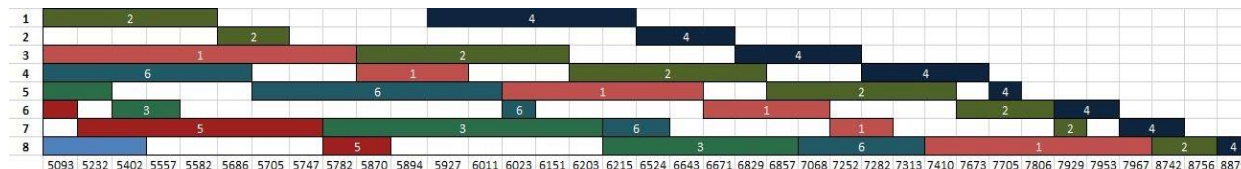
<i>car1.csv</i>		MESIN				
		1	2	3	4	5
PEKERJAAN	1	375	12	142	245	412
	2	632	452	758	278	398
	3	12	876	124	534	765
	4	460	542	523	120	499
	5	528	101	789	124	999
	6	796	245	632	375	123
	7	532	230	543	896	452
	8	14	124	214	543	785
	9	257	527	753	210	463
	10	896	896	214	258	259
	11	532	302	501	765	988

Tabel Lampiran D2 waktu proses *dataset car8.csv*

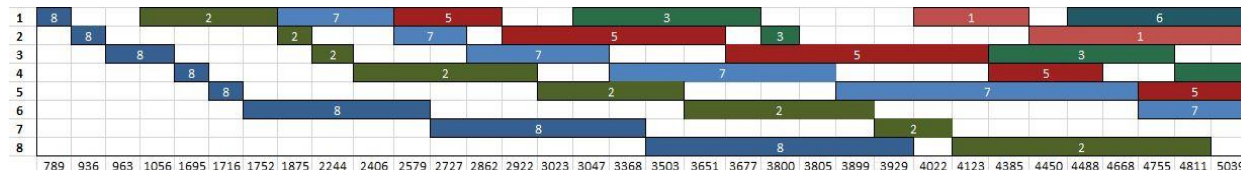
<i>car8.csv</i>		MESIN							
		1	2	3	4	5	6	7	8
PEKERJAAN	1	456	654	852	145	632	425	214	654
	2	789	123	369	678	581	396	123	789
	3	654	123	632	965	475	325	456	654
	4	321	456	581	421	32	147	789	123
	5	456	789	472	365	536	852	654	123
	6	789	654	586	824	325	12	321	456
	7	654	321	320	758	863	452	456	789
	8	789	147	120	639	21	863	789	654



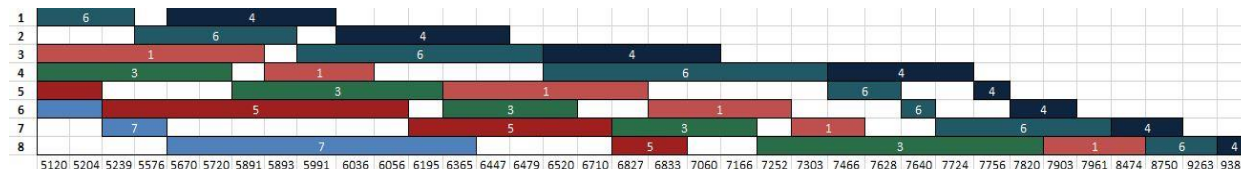
Gambar Lampiran D1 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *car8.csv* (Bagian 1)



Gambar Lampiran D2 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *car8.csv* (Bagian 2)



Gambar Lampiran D3 Hasil Uji Validitas *Gantt Chart* Metode GA *car8.csv* (Bagian 1)



Tabel Lampiran D3 waktu proses *dataset rec01.csv*

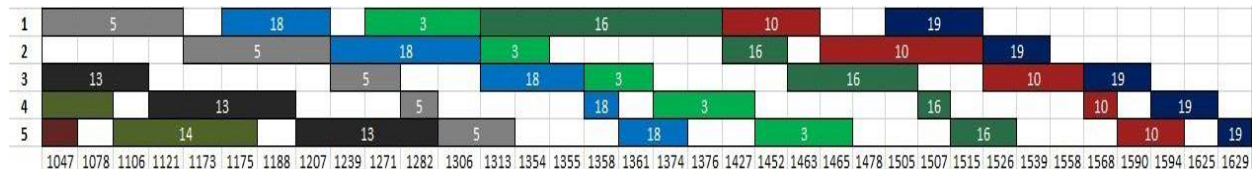
<i>rec01.csv</i>		MESIN										
		1	2	3	4	5		1	2	3	4	5
PEKERJAAN	1	5	76	74	99	26	11	25	59	88	87	40
	2	74	21	83	52	90	12	50	42	72	77	29
	3	67	48	6	66	38	13	58	76	71	82	94
	4	97	36	71	68	81	14	79	48	20	63	97
	5	87	86	64	11	31	15	35	57	78	99	80
	6	1	42	20	90	23	16	70	76	53	2	19
	7	69	32	99	26	57	17	79	22	77	74	95
	8	69	12	54	80	16	18	34	99	49	3	61
	9	11	63	24	16	89	19	37	24	32	35	4
	10	87	52	43	10	26	20	50	88	46	63	76



Gambar Lampiran D5 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec01.csv* (Bagian 1)



Gambar Lampiran D6 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec01.csv* (Bagian 2)



Gambar Lampiran D7 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec01.csv* (Bagian 3)



Gambar Lampiran D8 Hasil Uji Validitas *Gantt Chart* Metode GA *rec01.csv* (Bagian 1)



Gambar Lampiran D9 Hasil Uji Validitas *Gantt Chart* Metode GA *rec01.csv* (Bagian 2)

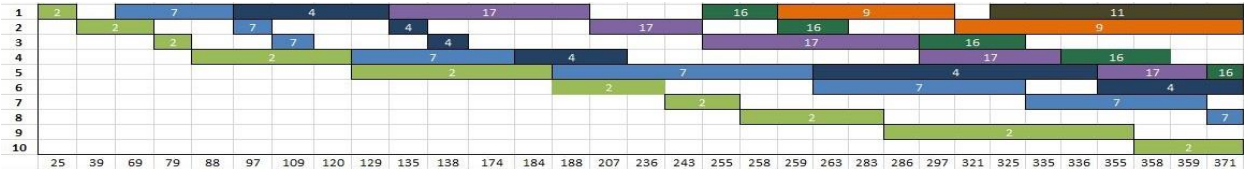


Tabel Lampiran D4 waktu proses *dataset rec11.csv* (Bagian 1)

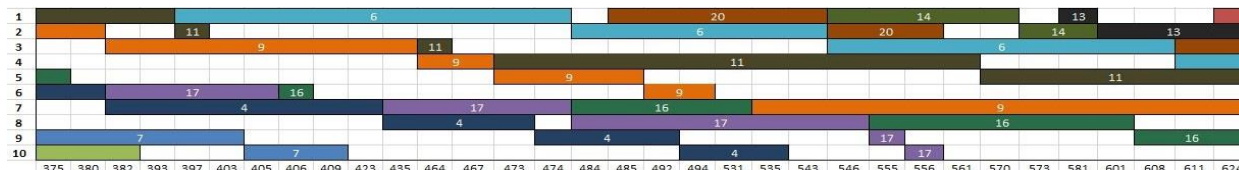
<i>rec11.csv</i>		MESIN									
		1	2	3	4	5	6	7	8	9	10
PEKERJAAN	1	96	80	56	48	14	88	50	15	67	65
	2	25	44	10	41	64	52	19	28	72	27
	3	76	62	48	54	47	35	72	54	27	56
	4	41	6	3	33	77	44	43	50	19	43
	5	65	91	75	30	47	55	51	1	36	73
	6	81	69	65	93	61	3	44	17	6	14
	7	49	9	12	54	75	66	34	12	32	6
	8	93	89	31	14	37	57	33	96	32	45
	9	39	83	55	32	18	9	93	65	75	73
	10	52	46	64	13	54	62	45	80	19	64
	11	72	4	29	94	85	51	29	65	50	16
	12	55	43	47	32	87	97	41	86	17	30
	13	8	91	81	93	14	86	64	42	70	3
	14	27	11	94	38	33	67	8	55	99	18
	15	34	86	87	10	64	30	47	51	69	26

Tabel Lampiran D5 waktu proses *dataset rec11.csv* (Bagian 2)

<i>rec11.csv</i>		MESIN									
		1	2	3	4	5	6	7	8	9	10
PEKERJAAN	16	15	5	39	23	16	1	57	55	62	35
	17	59	55	43	49	23	25	51	72	9	1
	18	93	4	43	5	84	55	22	78	31	11
	19	20	91	73	41	100	38	75	9	76	71
	20	59	13	93	26	11	7	66	42	54	99



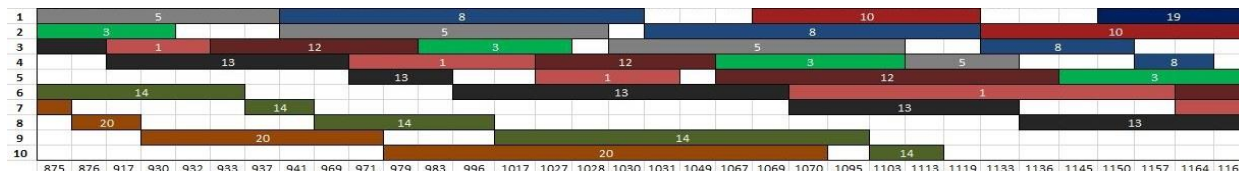
Gambar Lampiran D11 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec11.csv* (Bagian 1)



Gambar Lampiran D12 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec11.csv* (Bagian 2)



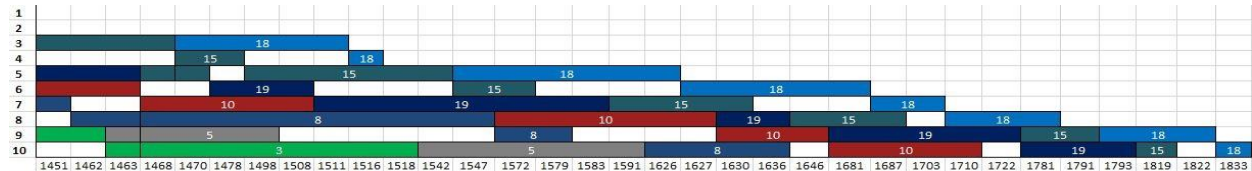
Gambar Lampiran D13 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec11.csv* (Bagian 3)



Gambar Lampiran D14 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec11.csv* (Bagian 4)



Gambar Lampiran D14 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec11.csv* (Bagian 5)



Gambar Lampiran D15 Hasil Uji Validitas *Gantt Chart* Metode GA-TS *rec11.csv* (Bagian 6)

BIODATA PENULIS



Ari Mohamad Barkhah lahir di Jakarta pada tanggal 22 April 1992. Penulis merupakan anak keempat dari empat bersaudara. Penulis menempuh pendidikan formal dimulai dari TK Mutiara Indonesia (1996-1998), SD Islam Al-Azhar 13 Rawamangun (1998-2004), SMP Islam Al-Azhar 12 Rawamangun (2004-2007), SMA Negeri 21 Jakarta (2007-2010) dan S1 Teknik Informatika ITS (2010-2015).

Bidang studi yang diambil oleh penulis pada saat kuliah di di Teknik Informatika ITS adalah Dasar dan Terapan Komputasi (DTK). Di luar bidang akademik, penulis suka membaca, mendengarkan musik, dan berolahraga. Penulis dapat dihubungi melalui email: ari.barkhah@gmail.com